General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

9950-

RF MODEL OF THE DISTRIBUTION SYSTEM AS A COMMUNICATION CHANNEL

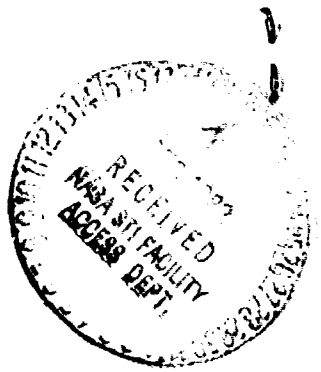
PHASE II

VOLUME IV — SOFTWARE SOURCE PROGRAM AND ILLUSTRATIVE ASCII DATABASE LISTINGS

FINAL REPORT
Contract No. 955647

July 28, 1982

R.C. Rustay
J.T. Gajjar
R.W. Rankin
R.C. Wentz
R. Wooding



General Electric Company
Corporate Research and Development
Schenectady, New York 12345

Prepared for

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91103

SRD-82-055-4

RF MODEL OF THE DISTRIBUTION SYSTEM AS A COMMUNICATION CHANNEL

PHASE II

VOLUME IV — SOFTWARE SOURCE PROGRAM AND ILLUSTRATIVE ASCII DATABASE LISTINGS

FINAL REPORT
Contract No. 955647

July 28, 1982

R.C. Rustay
J.T. Gajjar
R.W. Rankin
R.C. Wentz
R. Wooding

General Electric Company
Corporate Research and Development
Schenectady, New York 12345

Prepared for

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91103

This work was performed for the Jet Propulsion Laboratory, California Institute of Technology sponsored by the U.S. Department of Energy, through an agreement with the National Aeronautics and Space Administration.

SRD-82-055-4

ABSTRACT

This four-volume final report is concerned with Phase II of the DOE/JPL project "RF Model of the Distribution System As a Communication Channel." An earlier Phase I effort was concerned with the design, implementation, and verification of a computerized model for predicting the steady-state sinusoidal response of radial (tree) configured distribution feeders. That work demonstrated the feasibility and validity based on verification measurements made on a limited size portion of an actual live feeder. The Phase II effort is concerned with 1) extending the verification based on a greater variety of situations and network size, 2) extending the model capabilities for reverse direction propagation, 3) investigating parameter sensitivities, 4) improving transformer models, and 5) investigating procedures/fixes for ameliorating propagation "trouble spots."

PREFACE

THIS VOLUME CONTAINS LISTINGS OF SOURCE PROGRAMS AND SOME ILLUSTRATIVE EXAMPLES OF VARIOUS ASCII DATA BASE FILES. THE LISTINGS ARE GROUPED INTO THE FOLLOWING CATEGORIES:

MAIN PROGRAMS

SUBROUTINE PROGRAMS

ILLUSTRATIVE ASCII DATA BASE FILES

WITHIN EACH CATEGORY, FILES ARE LISTED ALPHABETICALLY.

TABLE OF CONTENTS

1. MAIN PROGRAM LISTINGS

```
AFPRYGEN                                                    04/16/81    2:56 PM

010 *       THIS PROGRAM GENERATES AFPRYLIJ TYPE FORMATTED FILES
020 *       TAILORED TO PSEUDO LOADS INVOLVING THREE PHASE TRANSFORMERS
030 *       AND THREE PHASE OPEN DELTA TRANSFORMERS WHOSE ADMITTANCE
040 *       MATRIX IS TWO BY TWO.  NOTE THAT THE USER WILL HAVE TO
050 *       RENAME OUFILE TO PROPER AFPRYLIJ NAME
060 *       NOTE THAT AS THIS PROGRAM NOW STANDS IT DOES NOT TREAT NECO=2
070 *       AS A 2X2 MATRIX, RATHER AS A 3X3 ASSOCIATED BY AN OPEN DELTA
080 *       IMPLIED BY NECO=2
090 *       THIS PROGRAM IS TO BE CONSIDERED TEMPORARY AND NOT
100 *       POLISHED.
110 *       NOTE THAT THE ADMITTANCE MATRICES ARE TESTED
120 *       FOR PHYSICAL REALIZABILITY (POSITIVE REAL TEST)
130 *       WITH:
140 *         ISTAT=N20  2X2  DETERMINET NEGATIVE
150 *         ISTAT=3NN  3X3  DETERMINET NEGATIVE
160 *         RATIO=(2*BR+AR)/AR
170     10 FORMAT(V)
180     11 FORMAT(I4,3X,3(1PE9.2,3X))
190     12 FORMAT(I4,2I5,4X,'0',I5,4X,'0',4X,'0')
200     13 FORMAT('1PRL,NECO,ISTAT,AR,BR,RATIO=',I4,I3,I5,1P3E10.2)
210        CHARACTER INFILE*9, OUFILE*9
220        DATA Z/0./
230        CALL NASTRK
240     50 PRINT,'ENTER "INFILE:" AND "OUFILE:"'
250        READ,INFILE,OUFILE
260        CALL OPENF(21,INFILE)
270        CALL OPENF(22,OUFILE)
280        WRITE(22,11)1000
290        WRITE(22,11)1010
300        LINE=1020
310    100 CONTINUE
320        ISTAT=0
330        READ(21,10,END=900)IPRL,NECO,AR,AI,BR,BI
340        WRITE(22,12)LINE,IPRL,3,NECO
350        LINE=LINE+10
360        WRITE(22,11)LINE
370        LINE=LINE+10
380        WRITE(22,11)LINE,AR
390        LINE=LINE+10
400        WRITE(22,11)LINE,AI
410        LINE=LINE+10
420        WRITE(22,11)LINE
430        LINE=LINE+10
440        IF(AR.LT.Z)ISTAT=1
450        IF((AR*AR-BR*BR).LT.Z)ISTAT=ISTAT+20
460        IF(NECO.EQ.3)GOTO200
470        IF(NECO.NE.2)STOP
480 *        LOGIC FOR NECO=2
490        WRITE(22,11)LINE,BR,AR
```

This program is provided as a convenience for generating AFPRYijk files when the 3X3 matrix admittance is balanced. See file R1 which has been transferred by this program into file R2.

```
500         LINE=LINE+10
510         WRITE(22,11)LINE,BI,AI
520         LINE=LINE+10
530         WRITE(22,11)LINE
540         LINE=LINE+10
550         WRITE(22,11)LINE,Z,Z,Z
560         LINE=LINE+10
570         WRITE(22,11)LINE,Z,Z,Z
580         GOTO300
590    200  CONTINUE
600  *        LOGIC FOR NECO=3
610         WRITE(22,11)LINE,BR,AR
620         LINE=LINE+10
630         WRITE(22,11)LINE,BI,AI
640         LINE=LINE+10
650         WRITE(22,11)LINE
660         LINE=LINE+10
670         WRITE(22,11)LINE,BR,BR,AR
680         LINE=LINE+10
690         WRITE(22,11)LINE,BI,BI,AI
700         IF((AR**3+BR*BR*(2.*BR-3.*AR)).LT.Z)ISTAT=ISTAT+300
710    300  LINE=LINE+10
720         WRITE(22,11)LINE
730         LINE=LINE+10
740         IF(ISTAT.GT.0)PRINT 13,IPRL,NECO,ISTAT,AR,BR,(2.*BR+AR)/AR
750         GOTO100
760    900  PRINT,'EOF REACHED ON ',INFILE
770         BACKSPACE 22
780         ENDFILE 22
790         PRINT,'OUTPUT FILE IS CALLED ',OUFILE
800         PRINT, 'TYPE CR TO CONTINUE'
810         READ,I
820         CALL DETACH(21,ISTAT,)
830         CALL DETACH(22,ISTAT,)
840         IF(I.EQ.0)GOTO50
850         PRINT,'RATIO=(2*BR+AR)/AR'
860         STOP
870         END
```

```
**********************************************************************

DNWKIN51                                          04/08/81     3:00 PM

**********************************************************************


010 * THIS Q&D PROGRAM COMPUTES THE TOTAL NUMBER OF EACH TYPE OF
020 * DISTRIBUTION TRANSFORMER FOUND IN A NETWORK FILE
030 * (IN THIS CASE DNWKIN56).  ALSO IT COMPUTES THE TOTAL
040 * CONNECTED DT LOAD (FOR NTTY EQUAL TO KVA) AND THE
050 * TOTAL CUMMULATIVE LENGTH OF THE FEEDER IN METERS*10
060 10 FORMAT(V)
070 CALL OPENF(11,'DNWKIN56;')
080 READ(11,10)N1,N2
090 200 READ(11,10,END=999)N1,N2,N3,N4,N5,N6,N7,N8,NTTY
100 IF(NTTY.EQ.05)I05=I05+1
110 IF(NTTY.EQ.10)I10=I10+1
120 IF(NTTY.EQ.15)I15=I15+1
130 IF(NTTY.EQ.25)I25=I25+1
140 IF(NTTY.EQ.40)I40=I40+1
150 IF(NTTY.EQ.50)I50=I50+1
160 NTOT=NTOT+NTTY
170 NLEN=NLEN+N7
180 GOTO200
190 999 CONTINUE
200 PRINT,'I05=',I05
210 PRINT,'I10=',I10
220 PRINT,'I15=',I15
230 PRINT,'I25=',I25
240 PRINT,'I40=',I40
250 PRINT,'I50=',I50
260 PRINT,'NTOT=',NTOT
270 PRINT,'NLEN=',NLEN
280 STOP
290 END
```

```
0010*# RUNH *=FEEDPUH5(ULIB,CORE=30K)LIBRARY4;
0020*#LIBRARY/IMSL,R;LIBRARY/OLDTSLIB,R
0030 *        THIS VERSION DIFFERS FROM FEEDPUS1 IN THAT:
0040 *          1)USER IS GIVEN OPTIONAL CHOICE FOR SELECTING IASM CODES
0050 *          2)ELIMINATES FIRST OF DATA RECORD PAIR
0060 *          3)USES A DRIECTORY VECTOR,IDIR TO STORE RECORD
0070 *            NUMBER VS LTYP TO CONSERVE CORE AND SPEED GENERATION.
0080 *            SEE END OF LISTING FOR DESCRIPTION OF STRATEGY
0090 *          4)UPPER AND LOWER LIMIT TESTING IS MADE ON EACH ELEMENT
0100 *            OF THE Z AND Y MATRICES.  NOTE THIS MODIFICATION
0110 *            ASSUMES THAT LTYP.GT.900 IMPLIES UG CABLE
0120 *          5)INCLUDES READING FROM ZYDAFILES FREQUENCY FREQK IN KHERTZ
0130 *            AND WRITES IT INTO DPUFILE HEADER RECORDS
0140 *          6)INCLUDES ABSOLUTE AND RELATIVE MODAL PROPAGATION
0150 *            SPEED CALCULATION IN OPTION 4
0151 *          7)PROVIDES USER WITH OPTION ON MAXIMUM FILE SIZE
0152 *          8)ELIMINATES MANY UNUSED VARIABLES IN DPULIJKN FILES
0153 *          9)USES LREC (RECORD SIZE) TO AUTOMATICALLY SELECT
0154 *            APPROPRIATE NUMBER OF RECORDS TO CONTAIN THE
0155 *            DIRECTORY ARRAY IDIR.  NRCD IS THE LAST RECORD
0156 *            USED FOR THIS PURPOSE.
0157 *         10)EXPECTS SYSTEM TO GROW AS NECESSARY.  RANDOM
0158 *            BINARY FILE NOT TO EXCEED MAXIMUM USER SPECIFIED
0159 *            LIMIT.
0160 *        NOTE THAT ANY SUBSEQUENT REVISIONS SHOULD NOT RESEQUENCE
0170 *        SO THAT COMPARE OPERATIONS MAY BE USED TO DETERMINE DIFFERENCE
0180 *        THIS MAIN PROGRAM CONSTRUCTS, MAINTAINS AND IF DESIRED LISTS
0190 *        THE DATABASE PER UNIT FILE DPULIJKN WHERE
0200 *          LI=IFRE= FREQUENCY CODE
0210 *           J=ITMP=TEMPERATURE CODE
0220 *           K=IRHO=EARTH CONDUCTIVITY CODE
0230 *           N=IASM=ASSUMPTION CODE
0240 *        NOTE ON DPULIJKN DATA RECORD FORMAT
0250 *          LTYP EVEN DENOTES FEEDER+SECONDARY CONDUCTORS
0260 *          LTYP ODD  DENOTES FEEDER ONLY.
0270 *          IT WILL BE REQUIRED THAT NUMS=0=NEUS FOR LTYP ODD.
0280 *        POLICY ON REVISIONS:
0290 *          ANY DATA RECORD WITH INVALID DATA WILL BE DENOTED BY LTYP.LT.0
0300 *          ANY DATA ON INPUT FILE WHICH IS TOO REPLACE GIVEN LTYP DATA
0310 *            MUST HAVE
0320 *                1) INTERNAL IREV=1  FROM TERMINAL
0330 *                2) INPUT IRVI=1.OR.DPU LTYP.LE.0
0340 *          NOTE THAT A DPU LTYP=0 INDICATES NULL DATA
0350 *        POLICY ON IFRE, ITMP, IRHO SPECIFICATION
0360 *          SOMEWHAT ARBITRARILY WILL ASSUME THAT IFRE,ITMP,IRHO
0370 *          WILL ALWAYS BE SPECIFIED BY THE USER.  THE IMPLICATION
0380 *          IS THAT IOPT=2,3 ONLY THE PERTINET (CORRESPONDING
0390 *          TO IFRE,ITMP,IRHO) DATA WILL BE ACCEPTED FROM THE
0400 *          INPUT FILE
```

```
0410 *       NOTE THAT THIS PROGRAM USES NON SYSTEM FORMAT FOR RANDOM
0420 *       BINARY FILES BECAUSE THE CRDC 605 SYSTEM HAS SOFTWARE
0430 *       ERROR WHICH FOR SOME UNKNOWN REASON WILL NOT ALLOW
0440 *       STANDARD SYSTEM FORMAT
0450     10 FORMAT(V)
0460     11 FORMAT(I4,10I5)
0470     12 FORMAT('CHECK ON DEGREE OF ORTHOGONALITY OF S')
0480     13 FORMAT('LTYP=',2I4)
0490     14 FORMAT(A3,I2,3I1,A1)
0500     15 FORMAT('IFRE,ITMP,IRHO=',I4,2I5)
0510     16 FORMAT('TYPE CR IF OK')
0520     18 FORMAT('RELE DPU',I2,3I1,A1)
0530     19 FORMAT('LTYP INVALIDATED=',2I5)
0540     20 FORMAT('LTYP=',I3,'  IERR=',I4,'  Z(',I1,',',I1,')=',1P2E16.8)
0550     21 FORMAT('LTYP=',I3,'  IERR=',I4,'  Y(',I1,',',I1,')=',1P2E16.8)
0560     22 FORMAT(A8,A1)
0570     23 FORMAT('OUTPUT FILENAME=',A9)
0580     24 FORMAT('INPUT  FILENAME=',A9)
0590     25 FORMAT('IFRE=',I2,5X,'ITMP=',I1,5X,'IRHO=',I1,5X,'IASM=',I1,
0600       &       5X,'FREQ=',1PE13.7)
0610     26 FORMAT('IASM=',I2)
0620     27 FORMAT(A1)
0630     28 FORMAT('HEADER RECORD OF FILE ',A8/' IFRE ITMP IRHO IASM ',
0640       &       'NRCD LREC NTYP NDIM IREC   DATE    TIME     FREQUENCY')
0650     29 FORMAT(I4,4I5,I6,I5,I4,I5,A10,F7.2,1PE16.7)
0660     31 FORMAT('LTYP=',I5,5X,'NECO=',I1,5X,'NTOT=',I1,5X,'NUMF=',I1,
0670       &       5X,'NUMS=',I1,5X,'NEUS=',I1,5X,'NEUF=',I1)
0680     32 FORMAT('SUM OF EIGENVECTOR CURRENT COMPONENTS')
0690     33 FORMAT('COULD NOT DETACH FILE',A10,' WITH STATUS =',I3)
0700     34 FORMAT('CHECK ON DEGREE OF ORTHOGANALITY OF Q')
0710     35 FORMAT('ASSUMPTION CODES,'5I2,' WILL BE PROCESSED')
0720     36 FORMAT('TYPE TOTAL NUMBER OF CODES FOLLOWED BY SINGLE DIGITS')
0730     37 FORMAT('NUMBER OF RECORDS=',I4,/'DIRECTORY CONTENTS')
0740     38 FORMAT(10(2I4,2X))
0750     39 FORMAT(A6,'=',1PE8.1,A8,'=',E8.1)
0760     40 FORMAT('LIMIT FILENAME=',A9)
0770     41 FORMAT('ZYDA',I2,2I1,';')
0780     42 FORMAT('LTYP ZRIIMI   ZRIIMA   ZIIIMI   ZIIIMA',
0790       &         '   ZRIJMI   ZRIJMA   ZIIJMI   ZIIJMA',
0800       &         '   YIIIMI   YIIIMA   YIIJMI   YIIJMA')
0810     43 FORMAT(I4,1P12E9.1)
0820     44 FORMAT('FREQUENCY IN HERTZ=',1PE16.7)
0821     45 FORMAT('DPU MAXIMUM SIZE=',I4,' BLOCKS CORRESPONDING TO ',
0822       &       'APPROXIMATELY',I5,' LTYPES')
0823     46 FORMAT('IF OKAY TYPE CR, ELSE TYPE DESIRED MAXIMUM SIZE IN BLOCKS')
0830        COMPLEX Z(9,9),ZO(9,9),S(9,9),SI(9,9),DL(9)
0840        COMPLEX Y(9,9),YO(9,9)
0850        COMPLEX SA1(9,9),SA2(9,9),SA3(9,9),SA4(9,9),SA5(9,9),SA6(9,9)
0860        COMPLEX ZERO,SV(9),SUM,ZE,YE
0870        DIMENSION IDIR(900),ITEMP(900)
0880        DIMENSION WKAREA(200),ICBUF(5),IFILEF(5),IASMV(5)
0890        CHARACTER DPU*3,FNAMS*9,INFILE*9,STRING*14,NAME*6,LIFILE*9
```

FEEDPUS5

```
0900        CHARACTER OUFILE*9,NDATE*8,CV*8,FEEDPU*6,FNAME*8
0910        DATA IASM/4/,IASMV/4,0,0,0,0/,LUINP/31/
0920        DATA IFRE/10/,IRHO/4/,ITMP/6/,LREC/210/,NRPT/1/,IASMAX/1/
0930        DATA DPU/'DPU'/
0940        DATA LEXI/9999/,IHUN/100/,ITHO/1000/,NDIM/9/,MDIM/900/
0950        DATA ITETE/1010/,NW/200/,IONT/120/
0960        DATA ICBUF/2,0,100,1,0/,IFILEF/5*0/,NAME/'FEEDPU'/
0970        DATA IOUT/0/,IINP/0/,OUFILE/'OUTFILE1;'/
0980        DATA LIFILE/'FEEDLIMT;'/
0990        DATA ZERO/(0.,0.)/,RZERO/0./,XBIG/1.E20/
1000        DATA ZOIIRL/3.E-3/,ZOIIRH/2.E-2/
1010        DATA ZOIIIL/4.E-2/,ZOIIIH/2.E-1/
1020        DATA ZOIJRL/4.E-3/,ZOIJRH/7.E-3/
1030        DATA ZOIJIL/1.E-2/,ZOIJIH/8.E-2/
1040        DATA ZUIIRL/4.E-3/,ZUIIRH/8.E-3/
1050        DATA ZUIIIL/4.E-2/,ZUIIIH/7.E-2/
1060        DATA ZUIJRL/4.E-3/,ZUIJRH/5.E-3/
1070        DATA ZUIJIL/4.E-2/,ZUIJIH/6.E-2/
1080        DATA YOIIIL/2.E-7/,YOIIIH/1.E-6/
1090        DATA YOIJIL/-7.E-7/,YOIJIH/-6.E-9/
1100        DATA TPI/6.2831851/,CLIGHT/2.997925E8/
1110 *        DEFINE ASF
1120        IFCF(I)=IASMV(I)+10
1150        CALL FPARAM(1,IONT)
1160        CALL DATIM(NDATE,TIME)
1180        ICBUF(2)=12*LREC/320
1200    100 PRINT 15,IFRE,ITMP,IRHO
1210        PRINT,'TYPE 1 IF INCORRECT,-1 TO EXIT OR CR IF OK'
1220        READ,I
1230        IF(I)7000,110,102
1240    102 PRINT 15
1250        READ,IFRE,ITMP,IRHO
1260        GOTO100
1270    110 CONTINUE
1280        CALL NASTRK
1290        ENCODE(INFILE,41)IFRE+1000,ITMP,IRHO
1300        CALL YASTRK
1310        IBUI=0
1320        ILP=1
1330 *        USER TYPES 0 IE CR TO ESCAPE
1340 *                   1 TO INITIATE LTYP RECORD INVALIDATION
1350 *                   2 TO BUILD/REBUILD FROM SCRATCH
1360 *                   3 ADD TO EXISTING FILES INCLUDING REVISION
1370 *                   4 TO PRINT DOCUMENTATION
1380        PRINT,'TYPE 0-ESCAPE,1-CANCEL,2-BUILD/REBUILD,3-ADD/REVISE,4-PRINT'
1390        READ,IOPT
1400        IF(IOPT.NE.2)GOTO190
1410        PRINT,'TYPE 1 FOR LIMIT CHECKING OTHERWISE A CR'
1420        READ,ILIMIT
1430        IF(ILIMIT.EQ.0)GOTO190
1440        PRINT,'TYPE 1 TO SPECIFY OUTPUT FILE TO CONTAIN LIMIT CHECK RESULTS
1450        PRINT,'OTHERWISE A CR FOR TERMINAL OUTPUT'
```

```
      READ,I
      IF(I.EQ.0)GOTO170
  120 PRINT 40,LIFILE
      PRINT 16
      READ,I
      IF(I.EQ.0)GOTO130
      PRINT 40
      READ,LIFILE
      GOTO120
  130 LULIMT=33
      CALL OPENF(LULIMT,LIFILE,ISTAT)
      IF(ISTAT.NE.0)CALL ERRSTT(NAME,130,LULIMT,ISTAT,0)
      PRINT,'OPENED FILE ',LIFILE
      GOTO180
  170 LULIMT=06
  180 CONTINUE
      WRITE(LULIMT,39)'ZOIIRL',ZOIIRL,'ZOIIRH',ZOIIRH
      WRITE(LULIMT,39)'ZOIIIL',ZOIIIL,'ZOIIIH',ZOIIIH
      WRITE(LULIMT,39)'ZOIJRL',ZOIJRL,'ZOIJRH',ZOIJRH
      WRITE(LULIMT,39)'ZOIJIL',ZOIJIL,'ZOIJIH',ZOIJIH
      WRITE(LULIMT,39)'ZUIIRL',ZUIIRL,'ZUIIRH',ZUIIRH
      WRITE(LULIMT,39)'ZUIIIL',ZUIIIL,'ZUIIIH',ZUIIIH
      WRITE(LULIMT,39)'ZUIJRL',ZUIJRL,'ZUIJRH',ZUIJRH
      WRITE(LULIMT,39)'ZUIJIL',ZUIJIL,'ZUIJIH',ZUIJIH
      WRITE(LULIMT,39)'YOIIIL',YOIIIL,'YOIIIH',YOIIIH
      WRITE(LULIMT,39)'YOIJIL',YOIJIL,'YOIJIH',YOIJIH
  190 CONTINUE
      IF(IOPT.LT.0.OR.IOPT.GT.4)GOTO110
      IF(IOPT.EQ.0)GOTO100
      IF(IOPT.EQ.4)GOTO202
      PRINT,'TYPE 1 FOR DIAGNOSTICS'
      READ,IDIA
      IF(IDIA.NE.1)GOTO210
  202 PRINT,'TYPE 1 FOR TERMINAL OUTPUT'
      READ,I
      IF(I.NE.0)GOTO210
      LUOUT=32
*       READ NAME FOR OUTPUT FILE
  203 PRINT 23,OUFILE
      PRINT 16
      READ,I
      IF(I.EQ.0)GOTO205
*       DETACH OUTPUT FILE IF OPEN AND NEW FILENAME BEING OPENED
      IF(IOUT.EQ.0)GOTO204
      CALL DETACH(LUOUT,ISTAT,)
      IF(ISTAT.NE.0)PRINT 33,OUFILE,ISTAT
      IOUT=0
  204 CONTINUE
      PRINT 23
      READ,FNAME
      ENCODE(OUFILE,22)FNAME,';'
      GOTO203
```

```
1970    205 CONTINUE
1980        IF(IOUT.EQ.1)GOTO220
1990        CALL OPENF(LUOUT,OUFILE,ISTAT)
2000        IF(ISTAT.NE.0)CALL ERRSTT(NAME,205,LUOUT,ISTAT,IOUT)
2010        PRINT,'OPENED OUTPUT FILE=',OUFILE
2020        IOUT=1
2030        GOTO220
2040    210 LUOUT=6
2050    220 CONTINUE
2060        IF(IINP.EQ.0)GOTO221
2070        CALL DETACH(LUINP,ISTAT,)
2080        IF(ISTAT.NE.0)PRINT 33,INFILE,ISTAT
2090        IINP=0
2100 *        DEFINE INPUT FILE FOR IOPT=2,3
2110    221 IF(IOPT.EQ.1.OR.IOPT.EQ.4)GOTO240
2120    222 PRINT 24,INFILE
2130        PRINT 16
2140        READ,I
2150        IF(I.EQ.0)GOTO225
2160        PRINT 24
2170        READ,FNAME
2180        ENCODE(INFILE,22)FNAME,';'
2190        GOTO222
2200    225 CONTINUE
2210        CALL OPENF(LUINP,INFILE,ISTAT,1,0,1)
2220        IF(ISTAT.NE.0)CALL ERRSTT(NAME,225,LUINP,ISTAT,0)
2230        IINP=1
2240    240 CONTINUE
2250 *        DEFINE ASSUMTION CODES FOR IOPT=1,2,3
2260        IF(IOPT.EQ.4)GOTO250
2270        PRINT 35,(IASMV(I),I=1,IASMAX)
2280        PRINT 16
2290        READ,I
2300        IF(I.EQ.0)GOTO250
2310        PRINT 36
2320        READ,IASMAX,(IASMV(I),I=1,IASMAX)
2330        GOTO240
2340 *        NOW THAT INFILE AND OUFILE HAVE BEEN SPECIFIED
2350 *        BRANCH TO APPROPRIATE OPERATION
2360    250 GOTO(1000,2000,3000,4000),IOPT
2370   1000 CONTINUE
2380 *        LOGIC FOR INVALIDATING/CANCELLING LTYP DATA RECORD PAIRS
2390 *        ADOPT POLICY TO MODIFY ALL POSSIBLE FILES, NOT THOSE
2400 *        WHICH ARE SPECIFIED AT THE MOMENT VIA IASMV(I)
2410 *        SET FLAG FOR LOGIC OF COMPARING NTYP
2420        INTYP=1
2430        DO 1076 I=1,IASMAX
2440        IFC=I+10
2450 *        IF ALREADY OPEN BYPASS CALL OPENF
2460        IF(IFILEF(I).GT.0)GOTO1076
2470 *        OPEN BUT DO NOT CREATE
2480        CALL NASTRK
```

```
FEEDPUSS

2490        ENCODE(FNAMS,14)'DPU',IFRE,1000,ITYP,IRHO,1,'.'
2500        CALL YASTRX
2520        CALL OPEN(IFC,FNAMS,ISTAT,4,1,1)
2540        IF(ISTAT.EQ.0)GOTO1070
2560        PRINT,'UNABLE TO OPEN DPUFILE,ISTAT=',FNAMS,ISTAT
2580        GOTO1076
2600  1070  PRINT,'OPENED ',FNAMS
2620        IFILEF(I)=1
2640        CALL RANSI2(IFC,LREC,1)
2660  *       NOW AS A LOGICAL CHECK READ HEADER AND COMPARE
2680        READ(IFC'1)I1,I2,I3,I4,I5,I6,I7,I8,I9,CV,TEMP,FREQ
2700        IF(I1.NE.IFRE.OR.I2.NE.ITYP.OR.I3.NE.IRHO.OR.I4.NE.
2720       &   1.OR.I6.NE.LREC.OR.I8.NE.NDIM)GOTO6000
2740        DO 1071 J=2,I5
2760        K1=(J-2)*LREC+1
2780        K2=MIN0(I7-99,(J-1)*LREC)
2800        READ(IFC'J) (ITEMP(K), K=K1,K2)
2820        IF(K2.EQ.I7-99)GOTO1072
2840  1071  CONTINUE
2860  1072  CONTINUE
2880        IF(INTYP.EQ.0)GOTO1074
2900        DO 1073 J=1,I7-99
2920  1073  IDIR(J)=ITEMP(J)
2940        IREC=I9
2960        NTYP=I7
2980        INTYP=0
3000        GOTO1076
3020  1074  IF(I9.NE.IREC) CALL ERRSTT(NAME,1074,I,I9,IREC)
3040        IF(I7.NE.NTYP) CALL ERRSTT(NAME,1075,I,I7,NTYP)
3060        DO 1075 J=1,NTYP-99
3080  1075  IF(ITEMP(J).NE.IDIR(J)) CALL ERRSTT(NAME,1076,I,J,ITEMP(J))
3100  1076  CONTINUE
3120        PRINT,'TYPE LTYP FOR DATA TO BE CANCELLED'
3140  1077  PRINT 13
3160        READ,LTYP
3180        IF(LTYP.EQ.0)GOTO1200
3200        IF(LTYP.LT.1HUN.OR.LTYP.GE.1THO)GOTO1077
3220        IF(LTYP.LE.NTYP)GOTO1078
3240        PRINT,'LTYP EXCEEDS NTYP=',NTYP
3260        GOTO1078
3280  *       AT THIS POINT READY TO CANCEL DATA
3300  1078  IR=IDIR(LTYP-99)
3320        DO 1100 I=1,5
3340        IF(IFILEF(I).EQ.0)GOTO1100
3360        IFC=I+10
3380        CALL DPUFRW(IFC,IR,0,NDIM,I2,N,NTOT,NUMF,NUMS,NEUS,NEUF,
3400       &             Z,Y,Z0,Y0,S,SI,DL)
3420        IF(I2.EQ.LTYP)GOTO1088
3440        IF(I2)1082,1084,1086
3460  1082  IF(I2.NE.(-LTYP))CALL ERRSTT(NAME,1082,I2,LTYP,I4)
3480  *       I2=-LTYP
3500        PRINT,'LTYP RECORD ALREADY CANCELLED,FILE',I2
```

```
2950        GOTO1090
2960  1085  PRINT,'NULL RECORD,FILE',I2,LTYP
2970        GOTO1090
2980  1086  PRINT,'DISCREPANCY IN FILE',I2,LTYP
2990        CALL ERRSTT(NAME,1086,I2,LTYP,I4)
3000  1088  I2=-I2
3010        CALL DPUFRW(IFC,IR,1,NDIM,I2,N,NTOT,NUMF,NUMS,NEUS,NEUF,
3020       &            Z,Y,ZO,YO,S,SI,DL)
3030  1090  CONTINUE
3040  1100  CONTINUE
3050        PRINT 19,LTYP
3060        GOTO1077
3070  1200  CONTINUE
3080  *       AT THIS POINT GIVE USER CHANCE TO EXIT FROM THIS
3090  *       PROCEDURE OF INVALIDATING LTYP RECORDS
3100        PRINT,'TYPE 1 TO CONTINUE INVALIDATION OF ANOTHER COMBINATION'
3110        READ,I
3120        IF(I)1000,110,1000
3130  2000  CONTINUE
3140  *       BUILD/REBUILD FROM SCRATCH
3150        DO 2200 I=1,5
3160  *       POLICY IS TO RELEASE ALL DPU FILES WITH THIS GENDER
3170        IFC=I+10
3180  *       MAKING SURE FOR ALL COMBINATIONS THAT FNAMS DEFINED
3190        CALL NASTRK
3200        ENCODE(FNAMS,14)'DPU',IFRE+1000,ITMP,IRHO,I,';'
3210        CALL YASTRK
3220  *       IF FILE IS OPEN GO TO RELEASE SEQUENCE
3230        IF(IFILEF(I).GT.0)GOTO2100
3240  *       OTHERWISE ATTEMPT TO FIRST OPEN WITHOUT CREATING
3260        CALL OPENF(IFC,FNAMS,ISTAT,3,1,1)
3270  *       IF SUCCESSFUL GO TO RELEASE SEQUENCE
3280        IF(ISTAT.EQ.0)GOTO2100
3290  *       IF ISTAT=5 FILE DOES NOT EXIST
3300        IF(ISTAT.EQ.5)GOTO2200
3310  *       OTHERWISE EXIT
3320        PRINT,'DPU FILE AND STATUS=',IFRE,ITMP,IRHO,I,ISTAT
3330        GOTO7000
3340  2100  CONTINUE
3350  *       RELEASE SEQUENCE
3360        CALL DETACH(IFC,ISTAT,)
3370        IF(ISTAT.EQ.0)GOTO2110
3380        PRINT,'UNABLE TO DETACH DPUFILE IASM,ISTAT=',I,ISTAT
3390        GOTO7000
3400  2110  IFILEF(I)=0
3410        CALL NASTRK
3420        ENCODE(STRING,18)IFRE+1000,ITMP,IRHO,I,'\'
3430        CALL YASTRK
3440        CALL CALLSS(STRING)
3450  2200  CONTINUE
3460  *       CREATE SEQUENCE
3461  2201  PRINT 45, ICBUF(3),(320*ICBUF(3)/LREC) - 2
```

```
3462        PRINT 46
3463        READ,I
3464        IF(I.EQ.0)GOTO2205
3465        ICBUF(3)=I
3466        GOTO2201
3467   2205 CONTINUE
3470        DO 2400 K=1,IASMAX
3480  *       CREATE ONLY DESIRED FILES
3490        I=IASMV(K)
3500        IFC=IASMV(K)+10
3520        CALL NASTRK
3530        ENCODE(FNAMS,14)'DPU',IFRE+1000,ITMP,IRHO,I,';'
3540        CALL YASTRK
3541  *       NOTE ICBUF(1)=2, IE. CREATING FILE ONLY IF IT DOES NOT EXIST
3550        CALL OPENF(IFC,FNAMS,ISTAT,3,1,ICBUF)
3560        IF(ISTAT.EQ.0)GOTO2210
3570        PRINT,'UNABLE TOO CREATE DPUFILE IASM,ISTAT=',I,ISTAT
3580        GOTO7000
3590   2210 CONTINUE
3600        IFILEF(I)=1
3610        PRINT,'OPENED ',FNAMS
3620        CALL RANSIZ(IFC,LREC,1)
3630   2400 CONTINUE
3640        NTYP=99
3641        NRCD=2+(MDIM+1)/LREC
3650        IREC=NRCD+1
3660        IBUI=1
3670        GOTO3408
3680   3000 CONTINUE
3690  *       ADD TO EXISTING FILES INCLUDING REVISIONS
3700  *       SET FLAG FOR LOGIC TO COMPARE NTYP
3710        INTYP=1
3720        DO 3200 I=1,5
3730  *       PICK ALL POSSIBLE FILES
3740        IFC=I+10
3750  *       IF ALREADY OPEN BYPASS CALL OPENF
3760        IF(IFILEF(I).GT.0)GOTO3122
3770  *       OPEN BUT DO NOT CREATE
3780        CALL NASTRK
3790        ENCODE(FNAMS,14)'DPU',IFRE+1000,ITMP,IRHO,I,';'
3800        CALL YASTRK
3820        CALL OPENF(IFC,FNAMS,ISTAT,3,1,1)
3830        IF(ISTAT.EQ.0)GOTO3120
3840        PRINT,'UNABLE TO OPEN DPUFILE,ISTAT=',FNAMS,ISTAT
3850  *       STRATEGICALLY WILL NOT ABORT SINCE FILE MAY NOT EXIST
3860        GOTO3200
3870   3120 CONTINUE
3880        IFILEF(I)=1
3890        PRINT,'OPENED ',FNAMS
3900        CALL RANSIZ(IFC,LREC,1)
3910  *       NOW AS A LOGICAL CHECK READ HEADER AND COMPARE
3920   3122 READ(IFC'1)I1,I2,I3,I4,I5,I6,I7,I8,I9,CV,TEMP,FREQ
```

```
3925        IF(I1.NE.IFRE.OR.I2.NE.ITMP.OR.I3.NE.IRHO.OR.I4.NE.
3930       &   I.OR.I6.NE.LREC.OR.I8.NE.NDIM)GOTO6000
3935        DO 3123 J=2,I5
3940        K1=(J-2)*LREC+1
3945        K2=MIN0(I7-99,(J-1)*LREC)
3950        READ(IFC'J) (ITEMP(K),K=K1,K2)
3955        IF(K2.EQ.I7-99)GOTO3124
3960   3123 CONTINUE
3965   3124 CONTINUE
3975        IF(INTYP.EQ.0)GOTO3140
3985        DO 3130 J=1,I7-99
3990   3130 IDIR(J)=ITEMP(J)
3995        IREC=I9
4000        NTYP=I7
4005        INTYP=0
4010        GOTO3200
4020   3140 IF(I9.NE.IREC) CALL ERRSTT(NAME,3140,I,I9,IREC)
4025        IF(I7.NE.NTYP) CALL ERRSTT(NAME,3141,I,I7,NTYP)
4030        DO 3150 J=1,NTYP-99
4035   3150 IF(ITEMP(J).NE.IDIR(J)) CALL ERRSTT(NAME,3150,I,J,ITEMP(J))
4050   3200 CONTINUE
4070        PRINT,'TYPE 1 TO ENABLE REVISIONS'
4080        READ,IREV
4090   3408 CONTINUE
4100 *        AT THIS POINT USE SAME LOGIC FOR IOPT=2,3 WITH IBUI
4110 *        FLAG AS CONTROL.  NOTE THAT NTYP HAS BEEN ESTABLISHED FOR
4120 *        EITHER CASE
4130        REWIND LUINP
4140   3410 READ(LUINP,10,END=3900)LINU,LTYP,IRVI,I1,I2,I3,NTOT,NUMF,
4150       &                           NUMS,NEUS,NEUF,FREQK
4160 *        NOTE FREQK AS READ IS IN KHERTZ. WILL CONVERT HERE
4170        FREQ=ITHO*FREQK
4180        IF(IDIA.NE.0.AND.ILP.EQ.1)WRITE(LUOUT,44)FREQ
4190 *        THE FOLLOWING LINU CHECK IS NOT NECESSARY
4200        IF(LINU.NE.ITETE)CALL ERRSTT(NAME,3410,ITETE,LINU,LUINP)
4210 *        IF I1,I2,I3 DO NOT MATCH SPECIFIED IFRE,IRHO,ITMP SEARCH FOR
4220 *        NEXT LINU=LEXI=9999
4230        IF(I1.EQ.IFRE.AND.I2.EQ.ITMP.AND.I3.EQ.IRHO)GOTO3430
4240   3412 READ(LUINP,10,END=9000)LINU
4250        IF(LINU.NE.LEXI)GOTO3412
4260        PRINT,'NO MATCH'
4270        GOTO3410
4280   3430 CONTINUE
4290 *        AT THIS POINT HAVE ACHIEVED A MATCH BETWEEN
4300 *        USER SPECIFIED IFRE,ITMP,IRHO AND SIMILAR
4310 *        QUATITIES READ FROM INFILE UPTO BUT NOT LTYP
4320        IF(LTYP.LT.IHUN.OR.LTYP.GE.ITHO)GOTO3480
4330 *        THE FOLLOWING ARE THE ONLY CONDITIONS FOR WHICH
4340 *        LTYP DATA WILL ALLOWED TO BE WRITTEN FROM
4350 *        INFILE TO DPUFILE.
4360        IF(IREV.EQ.1.AND.LTYP.LE.NTYP.AND.IRVI.EQ.1)GOTO3500
4370        IF(IREV.NE.1.AND.LTYP.GT.NTYP)GOTO3500
```

```
      IF(IBUI.EQ.1)GOTO3500
      IF(LTYP.GT.NTYP)GOTO3480
*       IREV-TERMINAL PERMISSION SEE LINE 2920
*       IRVI-READ FROM INPUT FILE
*       ALSO INCLUDED ARE THE FOLLOWING CONDITIONS
*       WHICH DEPEND ON READING DPUFILE TO GET ITS
*       LTYP.  WILL READ ONLY ONE, ASSUME OTHERS SAME
      READ(10+IASMV(1)*IDIR(LTYP-99))I2
      IF(I2.LE.0.AND.LTYP.LE.NTYP)GOTO3500
*       HAVING REJECTED THE CURRENT INPUT FILE DATA SE
*       SEARCH FOR THE BEGINNING OF A NEW SET
 3480 PRINT,'REJECTED SET WITH IRVI,LTYP,IFRE,ITMP,IRHO='
      PRINT 11,IRVI,LTYP,IFRE,ITMP,IRHO
      GOTO3412
 3500 CONTINUE
*       HAVE NOW DETERMINED THAT HAVE MET CONDITIONS FOR REPLACEMENT
*       OR ADDITION.
*       NOW READ Z AND Y
      I1=ITETE
      ZRIIMI= XBIG
      ZRIJMI= XBIG
      ZIIIMI= XBIG
      ZIIJMI= XBIG
      ZRIIMA=-XBIG
      ZRIJMA=-XBIG
      ZIIIMA=-XBIG
      ZIIJMA=-XBIG
      DO 3510 I=1,NTOT
      DO 3510 J=1,NTOT
      READ(LUINP,10)LINU,ZE
*       INSERTING LOGIC FOR LIMIT VALUES, NOTE LTYP.GT.900 IMPLIES CABLE
      ZR=REAL(ZE)
      ZI=AIMAG(ZE)
      IERR=0
      IF(I.EQ.J)GOTO3503
      ZRIJMI=AMIN1(ZRIJMI,ZR)
      ZRIJMA=AMAX1(ZRIJMA,ZR)
      ZIIJMI=AMIN1(ZIIJMI,ZI)
      ZIIJMA=AMAX1(ZIIJMA,ZI)
      IF(ILIMIT.EQ.0)GOTO3509
      IF(LTYP.GE.900)GOTO3505
      IF(ZR.LT.ZOIJRL.OR.ZR.GT.ZOIJRH)IERR=IERR+1
      IF(ZI.LT.ZOIJIL.OR.ZI.GT.ZOIJIH)IERR=IERR+10
      GOTO3508
 3503 CONTINUE
      ZRIIMI=AMIN1(ZRIIMI,ZR)
      ZRIIMA=AMAX1(ZRIIMA,ZR)
      ZIIIMI=AMIN1(ZIIIMI,ZI)
      ZIIIMA=AMAX1(ZIIIMA,ZI)
      IF(ILIMIT.EQ.0)GOTO3509
      IF(LTYP.GE.900)GOTO3507
      IF(ZR.LT.ZOIIRL.OR.ZR.GT.ZOIIRH)IERR=IERR+100
```

```
4900         IF(ZI.LT.ZOIIIL.OR.ZI.GT.ZOIIIH)IERR=IERR+1000
4910         GOTO3508
4920   3505  IF(ZR.LT.ZUIJRL.OR.ZR.GT.ZUIJRH)IERR=IERR+2
4930         IF(ZI.LT.ZUIJIL.OR.ZI.GT.ZUIJIH)IERR=IERR+20
4940         GOTO3508
4950   3507  IF(ZR.LT.ZUIIRL.OR.ZR.GT.ZUIIRH)IERR=IERR+200
4960         IF(ZI.LT.ZUIIIL.OR.ZI.GT.ZUIIIH)IERR=IERR+2000
4970   3508  IF(IERR.NE.0)WRITE(LULIMT,20)LTYP,IERR,I,J,ZE
4980   3509  Z(I,J)=ZE
4990         I1=I1+10
5000         IF(LINU.NE.I1)CALL ERRSTT(NAME,3510,LTYP,LINU,I1)
5010   3510  CONTINUE
5020         YIIIMI= XBIG
5030         YIIJMI= XBIG
5040         YIIIMA=-XBIG
5050         YIIJMA=-XBIG
5060         DO 3520 I=1,NTOT
5070         DO 3520 J=1,NTOT
5080         READ(LUINP,10)LINU,YE
5090         IERR=0
5100  *        INSERTING LOGIC FOR LIMIT VALUES, NOTE LTYP.GT.900 IMPLIES CABLE
5110         IF(REAL(YE).NE.RZERO)IERR=3000
5120         YI=AIMAG(YE)
5130         IF(I.EQ.J)GOTO3511
5140         YIIJMI=AMIN1(YIIJMI,YI)
5150         YIIJMA=AMAX1(YIIJMA,YI)
5160         IF(ILIMIT.EQ.0)GOTO3519
5170         IF(LTYP.GE.900)GOTO3512
5180         IF(YI.LT.YOIJIL.OR.YI.GT.YOIJIH)IERR=IERR+11
5190         GOTO3518
5200   3511  CONTINUE
5210         YIIIMI=AMIN1(YIIIMI,YI)
5220         YIIIMA=AMAX1(YIIIMA,YI)
5230         IF(ILIMIT.EQ.0)GOTO3519
5240         IF(LTYP.GE.900)GOTO3512
5250         IF(YI.LT.YOIIIL.OR.YI.GT.YOIIIH)IERR=IERR+12
5260         GOTO3518
5270   3512  IF(I.EQ.NTOT.AND.J.EQ.NTOT)GOTO3515
5280         IF(I.EQ.1.AND.J.EQ.1)YY=YI
5290         IF(I.EQ.J)GOTO3516
5300         IF(J.EQ.NTOT.OR.I.EQ.NTOT)GOTO3517
5310  *        ANY REMAING SHOULD BE ZERO
5320         IF(YI.NE.RZERO)IERR=IERR+13
5330         GOTO3518
5340   3515  IF(YI.LT.YY)IERR=IERR+14
5350         GOTO3518
5360   3516  IF(YI.NE.YY.OR.YI.LT.RZERO)IERR=IERR+15
5370         GOTO3518
5380   3517  IF(YI.NE.(-YY))IERR=IERR+16
5390   3518  IF(IERR.NE.0)WRITE (LULIMT,21)LTYP,IERR,I,J,YE
5400   3519  Y(I,J)=YE
5410         I1=I1+10
```

```
      IF(LINU.NE.I1)CALL ERRSTT(NAME,3520,LTYP,LINU,I1)
 3520 CONTINUE
*       NOW HAVE Z AND Y FOR NTOT CONDUCTORS INCLUDING SECONDARY
*       IF PRESENT. FOR SIMPLICITY OF DEBUGGING/CLARITY WILL TREAT
*       EACH DISTINCT COMBINATION SEPARATELY AT POSSIBLE EXPENSE OF
*       CODING AND COMPUTATION
*       NOW MAKE SOME TESTS
      IF(NEUF.NE.0.AND.NEUF.NE.1)GOTO8000
      IF(NEUS.NE.0.AND.NEUS.NE.1)GOTO8000
      IF(NTOT.LT.NUMF+NUMS+NEUS+NEUF)GOTO8000
      IF(LTYP.LT.IHUN.OR.LTYP.GE.ITHO)GOTO8000
      I2=MOD(LTYP,2)
      IF(I2.NE.0.AND.NUMS.NE.0)GOTO8000
      IF(I2.NE.0.AND.NEUS.NE.0)GOTO8000
      IF(I2.EQ.0.AND.NUMS.EQ.0)GOTO8000
      IF(NUMS.EQ.0.AND.NEUS.NE.0)GOTO8000
      IF(NUMS.NE.0.AND.NUMS.NE.2.AND.NUMS.NE.3)GOTO8000
*       PREPARE FOR WRITING BY FIRST CHECKING NTYP VS LTYP
*       FIRST CHECK TO SEE IF ATTEMPTING TO WRITE A
*       DUPLICATE LTYP. FORT IOPT=2 (IBUI=1) NEED ONLY
*       CHECK IDIR DIRECTORY VECTOR.  FOR IOPT=3
*       IN ADDITION NEED TO CHECK FOR NEGATIVE LTYP
*       IF IDIR INDICATES RECORD PRESENT.  AT A LATER
*       DATE CHANGE REVSION LOGIC SO THAT IDIR CONTAINS
*       INFORMATION
      IF(LTYP.GT.NTYP)GOTO3521
      IR=IDIR(LTYP-99)
      IF(IR.LE.0)GOTO3521
      READ(IASMV(1)+10'IR)I2
      IF(I2.GT.0)PRINT,'WRITING RECORD FOR DUPLICATE LTYP=',LTYP,I2
 3521 CONTINUE
      IF(LTYP.LT.NTYP)GOTO3524
      IF(LTYP.EQ.(NTYP+1))GOTO3523
      DO 3522 J=NTYP+1,LTYP-1
 3522 IDIR(J-99)=-1
 3523 NTYP=LTYP
 3524 CONTINUE
*       NOTE THAT IREC IS NEXT AVAILABLE DPU RECORD
*       BECAUSE MANY OPERATIONS ARE SAME WILL PERFORM IN A LOOP
*       USING SPECIAL LOGIC FOR EACH IASM
      DO 3600 I=1,5
*       USE IFILE(I) LOGIC INSTEAD OF IASMV(I) OPTION SINCE
*       WANT TO MODIFY ALL FILES NOT JUST THOSE IN IASMV(I)
      IF(IFILEF(I).EQ.0)GOTO3600
      IFC=I+10
      CALL CMTCOP(Z,SA1,NTOT,NDIM)
      CALL CMTCOP(Y,SA2,NTOT,NDIM)
      GOTO(3530,3540,3550,3560,3570),I
 3530 CONTINUE
*       LOGIC FOR IASM=1
      NECO=NUMF+NUMS+NEUF
*       IF SECONDARY AND FEEDER NEUTRALS ARE EACH PRESENT,
```

```
5940 *        INTERCHANGE FEEDER AND SECONDARY NEUTRAL, OTHERWISE ORDER OK
5950         IF(NEUS.EQ.0.OR.NEUF.EQ.0)GOTO3590
5960         K=NECO
5970         L=NECO+1
5980         GOTO3580
5990   3540  CONTINUE
6000 *        LOGIC FOR IASM=2
6010         NECO=NUMF+NEUF
6020 *        IF SECONDARIES AND FEEDER NEUTRAL BOTH PRESENT,
6030 *        INTERCHANGE FIRST SECONDARY AND FEEDER NEUTRAL,
6040 *        OTHERWISE OK
6050         IF(NUMS.EQ.0.OR.NEUF.EQ.0)GOTO3590
6060         K=NECO
6070         L=NUMF+NUMS+NEUS+1
6080         GOTO3580
6090   3550  CONTINUE
6100 *        LOGIC FOR IASM=3
6110         NECO=NUMF+NUMS
6120 *        NOTE Z,Y IN PROPER ORDER
6130         GOTO3590
6140   3560  CONTINUE
6150 *        LOGIC FOR IASM=4
6160         NECO=NUMF
6170 *        NOTE Z,Y IN PROPER ORDER
6180         GOTO3590
6190   3570  CONTINUE
6200 *        LOGIIC FOR IASM=5
6210         NECO=NUMF+NUMS+NEUS+NEUF
6220 *        NOTE Z,Y IN PROPER ORDER
6230         GOTO3590
6240   3580  CALL CMTCRC(Z,SA1,K,L,NTOT,NDIM)
6250         CALL CMTCRC(Y,SA2,K,L,NTOT,NDIM)
6260   3590  CALL CZYRED(SA1,SA2,SA3,SA4,NTOT,NECO,NDIM,NW,WKAREA)
6270 *        SA3,SA4 CONTAIN REDUCED Z,Y
6271         IF(IDIA.NE.0)WRITE(LUOUT,31)LTYP,NECO,NTOT,NUMF,NUMS,NEUS,NEUF
6280         CALL LINEPU(SA3,SA4,SA1,ZO,YO,S,SI,
6290        &            DL,SV,SA5,SA6,WKAREA,NECO,NDIM,NW,
6300        &            IDIA,LUOUT)
6310         IF(IDIA.EQ.0)GOTO3598
6311         DO 3591 J=1,NECO
6312   3591  SV(J)=DL(J)/FREQ
6313         WRITE(LUOUT,10)'DL/FREQ'
6314         CALL CVEPRT(SV,NECO,NDIM,LUOUT)
6315         WRITE(LUOUT,10)'YO'
6316         CALL CMTPRT(YO,NECO,NDIM,LUOUT)
6317 *        THE ABOVE OUTPUT PROVIDED FOR THE PURPOSE
6318 *        OF DETERMINING THE EFFECT OF FREQUENCY.
6320 *        CHECK DEGREE OF ORTHOGANALITY BY
6330 *       STRANSPOSE*SCONJUGATE
6340         CALL CMTABC(S,S,SA5,NECO,NECO,NECO,NDIM)
6350         WRITE(LUOUT,12)
6360         CALL CMTPRT(SA5,NECO,NDIM,LUOUT)
```

```
      WRITE(LUOUT,34)
      CALL CMTPRT(SA5,NECO,NDIM,LUOUT)
*       CALCULATE SUM OF CURRENT EIGENVECTOR COMPONENTS
      WRITE(LUOUT,32)
*       SINCE Q=TRANSPOSE OF SI, THE FOLLOWING CODE
*       CALCULATES THE SUM OF THE COLUMNS OF Q BY SUMMING
*       THE ROWS OF SI
      DO 3593 J=1,NECO
      SUM=ZERO
      DO 3592 K=1,NECO
 3592 SUM=SUM+SI(J,K)
 3593 SV(J)=SUM
      CALL CVEPRT(SV,NECO,NDIM,LUOUT)
      DO 3597 J=1,NECO
      X=TPI*FREQ/AIMAG(DL(J))
 3597 SV(J)=CMPLX(X,X/CLIGHT)
      WRITE(LUOUT,10)'ABSOLUTE AND RELATIVE SPEED'
      CALL CVEPRT(SV,NECO,NDIM,LUOUT)
 3598 CONTINUE
      LENREC=7+(12*NECO**2)+(2*NECO)
      IF(LENREC.GT.LREC) CALL ERRSTT(NAME,3598,NECO,LREC,LENREC)
      CALL DPUFRW(IFC,IREC,1,NDIM,LTYP,NECO,NTOT,NUMF,NUMS,NEUS,NEUF,
     &            SA3,SA4,ZO,YO,S,SI,DL)
 3600 CONTINUE
      IDIR(LTYP-99)=IREC
      IREC=IREC+1
      IF(ILP.EQ.0)GOTO3610
      PRINT 44,FREQ
      PRINT 42
      ILP=0
 3610 CONTINUE
      PRINT 43,LTYP,ZRIIMI,ZRIIMA,ZIIIMI,ZIIIMA,
     &            ZRIJMI,ZRIJMA,ZIIJMI,ZIIJMA,
     &            YIIIMI,YIIIMA,YIIJMI,YIIJMA
*       FINISHED WITH LTYP RECORD
*       NOW RETURN TO INPUT FILE WHOSE NEXT LINE NUMBER SHOULD
*       BE LEXI
      READ(LUINP,10)LINU
      IF(LINU.NE.LEXI)CALL ERRSTT(NAME,3900,LINU,LUINP,0)
      GOTO3410
 3900 CONTINUE
*       NOW THAT INPUT FILE IS EXHAUSTED AND NTYP ESTABLISHED
*       WRITE HEADERS
      DO 3910 I=1,5
      IF(IFILEF(I).EQ.0)GOTO3910
      IFC=I+10
      WRITE(IFC'1)IFRE,ITMP,IRHO,I,NRCD,LREC,NTYP,NDIM,IREC,NDATE,
     &      TIME,FREQ
      DO 3905 J=2,NRCD
      K1=(J-2)*LREC+1
      K2=MIN0(NTYP-99,(J-1)*LREC)
      WRITE(IFC'J) (IDIR(K),K=K1,K2)
```

```
6796        IF(K2.EQ.NTYP-99)GOTO3906
6797   3905 CONTINUE
6798  *
6799   3906 CONTINUE
6810   3910 CONTINUE
6820        PRINT,'EXHAUSTED INPUT FILE'
6830        IF(ILIMIT.NE.0.AND.LULIMT.NE.6)
6840       &      PRINT,'REMEMBER TO EXAMINE CONTENTS OF FILE ',LIFILE
6850        GOTO100
6860   4000 CONTINUE
6870  *       LOGIC FOR FORMAL PRINT OUTPUT
6880   4020 PRINT 26,IASM
6890        PRINT,'TYPE CR IF OK, OTHWERWISE NEW IASM'
6900        READ,I
6910        IF(I.EQ.0)GOTO4040
6940        IF(I.GT.0.AND.I.LE.5)GOTO4030
6950        PRINT,'IASM=1,2,3,4,5'
6960        GOTO4020
6970   4030 IASM=I
6980        GOTO4020
6990   4040 CONTINUE
7000        IFC=IASM+10
7010        CALL NASTRK
7020        ENCODE(FNAMS,14)'DPU',IFRE+1000,ITMP,IRHO,IASM,';'
7030        ENCODE(FNAME,14)'DPU',IFRE+1000,ITMP,IRHO,IASM
7040        CALL YASTRK
7050  *       IF ALREADY OPEN POLICY WILL BE TO DETACH
7060  *       SO THAT CAN EXAMINE ANOTHER FILE DURING SAME TRANSACTION
7070        IF(IFILEF(IASM).EQ.0)GOTO4042
7080        CALL DETACH(IFC,ISTAT,)
7090        IF(ISTAT.NE.0)PRINT,'LOGICAL ERROR AT 4040+7'
7100        IFILEF(IASM)=0
7110   4042 CONTINUE
7120  *       OPEN BUT DO NOT CREATE
7140        CALL OPENF(IFC,FNAMS,ISTAT,1,1,1)
7150        IF(ISTAT.EQ.0)GOTO4044
7160        PRINT,'UNABLE TO OPEN DPUFILE,ISTAT=',FNAMS,ISTAT
7170  *       GOTO EXIT SEQUENCE
7180        GOTO110
7190   4044 PRINT,'OPENED ',FNAMS
7200        IFILEF(IASM)=1
7210        CALL RANSIZ(IFC,LREC,1)
7220  *       NOW AS LOGIC CHECK READ HEADER AND COMPARE
7230        READ(IFC'1)I1,I2,I3,I4,NRCD,I6,NTYP,I8,IREC,CV,TEMP,FREQ
7231        IF(I1.NE.IFRE.OR.I2.NE.ITMP.OR.I3.NE.IRHO.OR.I4.NE.
7232       &   IASM.OR.I6.NE.LREC.OR.I8.NE.NDIM)GOTO6000
7233        DO 4046 J=2,NRCD
7234        K1=(J-2)*LREC+1
7235        K2=MIN0(NTYP-99,(J-1)*LREC)
7236        READ(IFC'J) (IDIR(K),K=K1,K2)
7237        IF(K2.EQ.NTYP-99)GOTO4047
7238   4046 CONTINUE
```

```
7240  4047 CONTINUE
7270       PRINT,'TYPE 100 FOR ENTIRE FILE'
7280       READ,I1
7290       IF(I1.NE.IHUN)GOTO4050
7300       I2=NTYP
7310       I3=1
7320       I4=1
7330       GOTO4060
7340  4050 CONTINUE
7350       PRINT,'TYPE 1 TO PRINT HEADER RECORD'
7360       READ,I3
7370       PRINT,'TYPE 1 TO PRINT DIRECTORY INFORMATION'
7380       READ,I4
7390  4055 PRINT,'TYPE TWO INTEGERS FOR SELECTED RANGE ON LTYP'
7400       READ,I1,I2
7410       IF(I1.LE.99)GOTO4055
7420       IF(I2.LE.NTYP)GOTO4056
7430       PRINT,'RESETTING MAXIMUM LTYP=NTYP'
7440       I2=NTYP
7450  4056 PRINT I3,I1,I2
7460       PRINT 16
7470       READ,I
7480       IF(I.NE.0)GOTO4055
7490  *      NOW HAVE ESTABLISHED ALL PRINT CONTROL
7500  *        I3=1 FOR HEADER
7510  *        I4=1 FOR DIRECTORY
7520  *        I1=FIRST LTYP
7530  *        I2=LAST LTYP
7540  *      NOT REWINDING LUOUT SINCE IT MAY CONTAIN DIAGNOSTICS
7550  4060 IF(I3.EQ.0)GOTO4070
7560       IF(LUOUT.EQ.06)CALL FORMFE
7570       WRITE(LUOUT,28)FNAME
7580       WRITE(LUOUT,29)IFRE,ITMP,IRHO,IASM,NRCD,LREC,NTYP,NDIM,IREC,
7590      6                    NDATE,TIME,FREQ
7600  4070 IF(I4.EQ.0)GOTO4100
7610       WRITE(LUOUT,37)IREC-1
7620       WRITE(LUOUT,38)(I+99,IDIR(I),I=1,NTYP-99)
7621  4100 IF(I2.LT.I1)GOTO4205
7630       DO 4200 I=I1,I2
7640       IR=IDIR(I-99)
7650       IF(IR.LE.0)GOTO4200
7660       IF(IR.GE.IREC)CALL ERRSTT(NAME,4100,I,IR,IREC)
7670       CALL DPUFRW(IFC,IR,0,NDIM,LTYP,N,NTOT,NUMF,NUMS,NEUS,NEUF,
7680      6            Z,Y,ZO,YO,S,SI,DL)
7690       IF(LTYP.EQ.0)GOTO4200
7700       IF(LUOUT.EQ.06)CALL FORMFE
7710       WRITE(LUOUT,22)FNAME
7720       WRITE(LUOUT,25)IFRE,ITMP,IRHO,IASM,FREQ
7730       WRITE(LUOUT,31)LTYP,N,NTOT,NUMF,NUMS,NEUS,NEUF
7740       WRITE(LUOUT,10)'Z'
7750       CALL CMTPRT(Z,N,NDIM,LUOUT)
7760       WRITE(LUOUT,10)'Y'
```

```
      CALL CMTPRT(Y,N,NDIM,LUOUT)
      WRITE(LUOUT,10)'ZO'
      CALL CMTPRT(ZO,N,NDIM,LUOUT)
      WRITE(LUOUT,10)'YO'
      CALL CMTPRT(YO,N,NDIM,LUOUT)
      WRITE(LUOUT,10)'S'
      CALL CMTPRT(S,N,NDIM,LUOUT)
      CALL CMTRAN(SI,SA1,N,N,NDIM)
      WRITE(LUOUT,10)'Q'
      CALL CMTPRT(SA1,N,NDIM,LUOUT)
*     WRITE(LUOUT,10)'SI'
*     CALL CMTPRT(SI,N,NDIM,LUOUT)
      WRITE(LUOUT,10)'DL'
      CALL CVEPRT(DL,N,NDIM,LUOUT)
*       COMPUTE RELATIVE AND ABSOLUTE MODAL PROPAGTION SPEEDS
      DO 4120 J=1,N
      X=TPI*FREQ/AIMAG(DL(J))
 4120 SV(J)=CMPLX(X,X/CLIGHT)
      WRITE(LUOUT,10)'ABSOLUTE AND RELATIVE SPEED'
      CALL CVEPRT(SV,N,NDIM,LUOUT)
      IF(LUOUT.EQ.06)CALL FORMFE
 4200 CONTINUE
 4205 IF(LUOUT.NE.06)PRINT,'WROTE ',OUFILE
      GOTO100
 6000 CONTINUE
*       ARRIVE HERE ON MISMATCH IN DPU HEADER PARAMETERS
      PRINT,'HEADER PARAMETERS NOT CONSISTANT'
      PRINT 11,IFRE,ITMP,IRHO,I,LREC,NDIM
      PRINT 11,I1,I2,I3,I4,I6,I8
      PRINT,'CHECK ALSO DIRECTORY VECTORS'
 7000 CONTINUE
*       TERMINATION DETACH PROCEDURES
      IF(IOUT.NE.0)CALL DETACH(LUOUT,ISTAT,)
      IF(ISTAT.NE.0)PRINT 33,OUFILE,ISTAT
      IF(IINP.NE.0)CALL DETACH(LUINP,ISTAT,)
      IF(ISTAT.NE.0)PRINT 33,INFILE,ISTAT
      DO 7010 I=1,IASMAX
      CALL DETACH(I+10,ISTAT,)
      IFILEF(I)=0
      IF(ISTAT.NE.0)PRINT,'UNABLE TO DETACH DPUFILE ISAM=',I
 7010 CONTINUE
      CALL ERRSTP(NAME,7010)
 8000 CONTINUE
*        REACH HERE WHEN ERROR IN INPUT FILE SPECIFICATION
*       OF LTYP,NTOT,NUMF,NUMS,NEUS,NEUF
      PRINT 11,LTYP,NTOT,NUMF,NUMS,NEUS,NEUF,I2
      CALL ERRSTT(NAME,8000,LTYP,NTOT,NECO)
 9000 CONTINUE
*       REACH HERE WHEN UNEXPECTED EOF OR LINE NUMBER IN INFILE
      CALL ERRSTT(NAME,9000,LINU,1,1)
      STOP
*       STRATEGY FOR MANAGING IDIR DIRECTORY VECTOR
```

```
FEEDPUS5

8400 *        A) NTYP IS THE MAXIMUM VALUE OF LTYP FOR WHICH A
8410 *           CORRESPONDING RECORD HAS BEEN WRITTEN, IE
8420 *           CAPABLE OF BEING READ
8430 *        B) SINCE LTYP.GE.100, LTYP-99 WILL BE USED TO ADDRESS
8440 *           DIRECTOR VECTOR IDIR, IE IDIR(LTYP-99)
8450 *        C) IF THE CONTENT OF ANY CELL (IN THE RANGE OF 1 TO
8460 *           NTYP-100) OF IDIR IS
8470 *            GT.0, THEN THE CELL CONTENT VALUE POINTS TO VALID RECORD
8480 *            LT.0, THEN THE DPU RECORD HAS NOT BEEN DEFINED AND CANNOT BE
8490 *        D) IREC IS THE NEXT AVAILABLE DPU RECORD
8500 *        E) THE STRUCTURE OF THE DPU FILES WILL BE
8510 *             RECORD # 1 - HEADER (INCLUDING NTYP AND NRCD)
8520 *             RECORD # 2 THRU # NRCD - CONTAIN DIRECTORY VECTOR IDIR
8521 *                      WITH NTYP-99 ENTRIES
8530 *             RECORD # NRCD+1 - FIRST OCCURING LTYP DATA RECORD
8540 *             RECORD # NRCD+2 - SECOND OCCURING LTYP DATA RECORD
8550 *               ETC.
8560       END
```

```
0010 *       THIS PROGRAM PROVIDES A PROCEDURE FOR GENERATING WITH MINIMAL
0020 *       INPUT, VARIOUS TYPES OF GENERIC NETWORKS RESTRICTED TO A
0030 *       MAXIMUM OF SINGLE LEVEL BRANCHING, ALL THREE PHASE, UNIFORM
0040 *       LINE TYPE IN EACH SEGMENT.  NORMALLY DISTRIBUTION TRANSFORMERS
0050 *       ARE ASSIGNED AT EACH NODE.  THIS OPTION CAN BE DEFEATED IF
0060 *       DESIRED.
0070 *       THE FOLLOWING IS A KEY TO THE COLUMN ASSIGNMENT IN ARRAY IA
0080 *         COL
0090 *           1  IPAR
0100 *           2  ILSN
0110 *           3  IRSN
0120 *           4  LTYP
0130 *           5  ILCD
0140 *           6  IOHM
0150 *           7  ITTY
0160 *           8  IPHS
0170 *           9  NSER
0180 *          10  IPRL
0190 *          11  IGD
0200 *          12  ACCUMULATED LENGTH
0210     10 FORMAT('TYPE CR IF OK')
0220     11 FORMAT(I5,5I5,1X,7I5,2I8)
0230     12 FORMAT('DNWKIN',I2,';')
0240     13 FORMAT('NETWORK IDENTIFICATION NUMBER=',I2)
0250     14 FORMAT('FOR TRUNK, LTYPM,ILENM,IDLEM=',3I7)
0260     15 FORMAT('FOR LATERAL',I2,' LTYP',I1,',ILEN',I1,',IDLE',I1,
0270      &        ',IDIS',I1,'=',4I7)
0280     16 FORMAT('NUMBER OF LATERALS=',I2)
0290        DIMENSION IA(2000,12),ILENL(10),IDISL(10),IDLEL(10),LTYPL(10)
0300        DIMENSION LENGT(2000)
0310        CHARACTER OUFILE *9
0320        DATA NID/10/,LU/12/,NLAT/2/
0330        DATA LTYPM/133/,ILENM/100000/,IDLEM/10000/
0340        DATA IOHM/4/,ITTY/25/
0350        DATA LTYPL/975,175,6*0/
0360        DATA ILENL/10000,5000,8*0/
0370        DATA IDISL/50000,70000,8*0/
0380        DATA IDLEL/500,500,8*0/
0390        CALL FPARAM(1,120)
0400    200 CONTINUE
0410        PRINT 13,NID
0420        PRINT 10
0430        READ,I
0440        IF(I)900,220,210
0450    210 PRINT 13
0460        READ,NID
0470        GOTO200
0480    220 CALL NASTRK
0490        ENCODE(OUFILE,12)NID+1000
```

```
0500        CALL YASTPK
0510        CALL OPENF(LU,OUFILE,ISTAT,2,0,1)
0520        IF(ISTAT.NE.5)GOTO225
0530        CALL OPENF(LU,OUFILE,ISTAT,2,0,0)
0540        IF(ISTAT.EQ.0)GOTO230
0550        PRINT,'UNABLE TO OPEN FILE ',OUFILE,' ISTAT=',ISTAT
0560        STOP 0210
0570   225  IF(ISTAT.NE.0)GOTO227
0580        PRINT,'FILE ',OUFILE,' ALREADY EXISTS,TYPE CR IF OK TO OVERWRITE'
0590        READ,I
0600        IF(I)200,230,200
0610   227  PRINT,OUFILE,' ISTAT=',ISTAT
0620        GOTO200
0630   230  PRINT 16,NLAT
0640        PRINT 10
0650        READ,I
0660        IF(I)200,230,235
0670   235  PRINT 16
0680        READ,NLAT
0690        IF(NLAT.GT.10)GOTO235
0700        GOTO230
0710   236  PRINT,'TYPE CR FOR AUTOMATIC DISTRIBUTION TRANSFORMER ASSIGNMENT'
0720        READ,ITFLAG
0730   240  CONTINUE
0740        PRINT 14,LTYPN,ILENN,IDLEN
0750        PRINT 10
0760        READ,I
0770        IF(I)200,260,250
0780   250  PRINT 14
0790        READ,LTYPN,ILENN,IDLEN
0800        GOTO240
0810   260  IF(NLAT.EQ.0)GOTO300
0820        DO 280 N=1,NLAT
0830   270  PRINT 15,N,N,N,N,N,LTYPL(N),ILENL(N),IDLEL(N),IDISL(N)
0840        PRINT 10
0850        READ,I
0860        IF(I.EQ.0)GOTO280
0870        PRINT 15,N,N,N,N,N
0871        READ,LTYPL(N),ILENL(N),IDLEL(N),IDISL(N)
0880        GOTO270
0890   280  CONTINUE
0900   300  CONTINUE
0910 *      NOW CONSTRUCT MAIN TRUNK ROUTE
0920        MLENG=0
0930        ISMAX=1
0940        IPHS=0
0950   310  CONTINUE
0960 *        DETERMINE LENGTH OF NEXT SECTION
0970        M=MIN0(IDLEN,ILENM-MLENG)
0980        MLENG=MLENG+M
0990        IPHS=MOD(IPHS,3)+1
1000 *        THE ABOVE GENERATES THE RECURSIVE SEQUENCE,1,2,3,1,2,3,ETC.
```

```
1010 *        FOR TRANSFORMER CONNECTION
1020          IA(ISMAX,1)=ISMAX-1
1030          IA(ISMAX,2)=0
1040          IA(ISMAX,3)=0
1050          IA(ISMAX,4)=LTYP
1060          IA(ISMAX,5)=M
1070          IA(ISMAX,6)=IOHM
1080          IA(ISMAX,7)=0
1090          IA(ISMAX,8)=0
1100          IF(ITFLAG.NE.0)GOTO315
1110          IA(ISMAX,7)=ITY
1120          IA(ISMAX,8)=IPHS
1130   315    IA(ISMAX,9)=0
1140          IA(ISMAX,10)=0
1150          IA(ISMAX,11)=50
1160          IA(ISMAX,12)=MLENG
1170 *        NOTE USING 12'TH COLUMN TO STORE ACCUMULATED LENGTH FROM SOURCE
1180 *        AT THIS POINT ISMAX=LAST SECTION NUMBER THUS FAR PROCESSED ON
1190 *        MAIN ROUTE
1200          IF(MLENG.EQ.ILENM)GOTO400
1210 *           PROCEED WITH NEXT MAIN LENGTH SECTION
1220          IA(ISMAX,2)=ISMAX+1
1240          ISMAX=ISMAX+1
1250          GOTO310
1260   400    CONTINUE
1270          ISMMM=ISMAX
1280          ILMMM=ILENM
1290 *        NOW WILL DO MASTER LOOP ON LATERALS IF ANY
1300          IF(NLAT.EQ.0)GOTO800
1310          DO 700 N=1,NLAT
1320 *        FIRST CHECK TO SEE IF LATERAL IS TO BE APPENDED TO END OF MAIN
1330 *        TRUNK OR IS TO BE ATTACHED AS A LATERAL ALONG MAIN ROUTE
1340 *        DEPENDING ON WHETHER IDISL(N) LARGER OR SMALLER THAN ILMMM
1350          I1=ILEPL(N)
1360          I2=IDLEL(N)
1370          I3=IDISL(N)
1380          IF(I3.LT.ILMMM)GOTO450
1390 *        NOW WANT TO APPEND THIS TO END OF CURRENT MAIN ROUTE
1391          IA(ISMAX,2)=ISMAX+1
1400          ISMAX=ISMAX+1
1410          MLENG=ILMMM
1420          ILMMM=ILMMM+I1
1430   410    CONTINUE
1440          M=MIN0(I2,ILMMM-MLENG)
1450          MLENG=MLENG+M
1460          IPHS=MOD(IPHS,3)+1
1470          IA(ISMAX,1)=ISMAX-1
1480          IA(ISMAX,2)=0
1490          IA(ISMAX,3)=0
1500          IA(ISMAX,4)=LTYPL(N)
1510          IA(ISMAX,5)=M
1520          IA(ISMAX,6)=IOHM
```

```
      IA(ISMAX,7)=0
      IA(ISMAX,8)=0
      IF(ITFLAG.NE.0)GOTO420
      IA(ISMAX,7)=ITTY
      IA(ISMAX,8)=IPHS
  420 IA(ISMAX,9)=0
      IA(ISMAX,10)=0
      IA(ISMAX,11)=50
      IA(ISMAX,12)=MLENG
      IF(MLENG.EQ.ILMIN)GOTO700
      IA(ISMAX,2)=ISMAX+1
      ISMAX=ISMAX+1
      GOTO410
  450 CONTINUE
*       AT THIS OPTION POINT THE LATERAL HAS TO BE ATTACHED TO SIDE
*       OF MAIN ROUTE.  NOW SEARCH FOR CORRECT DISTANCE
      DO 460 I=1,ISMAX
  460 IF(IA(I,12).GE.I3.AND.IA(I,3).EQ.0)GOTO470
*       IF LOGIC FALLS THRU THIS SEARCH IT MEANS THAT
*       PROBLEM HAS BEEN ILL DEFINED
      PRINT,'ISMAX,I3,IA(I,12)=',ISMAX,I3,IA(I,12)
*       NOW MODIFY IRSN IN SECTION I AND PLACE PARENT INFO IN IA
  470 ISMAX=ISMAX+1
      IA(I,3)=ISMAX
      IA(ISMAX,1)=I
*       NOW APPLY LOGIC
      MLENG=0
      IFLAG=1
      IPHS=MOD(IPHS,3)+1
  480 CONTINUE
      M=MINO(I2,I1-MLENG)
      MLENG=MLENG+M
      IPHS=MOD(IPHS,3)+1
      IF(IFLAG.EQ.0)IA(ISMAX,1)=ISMAX-1
      IA(ISMAX,2)=0
      IA(ISMAX,3)=0
      IA(ISMAX,4)=LTYPL(N)
      IA(ISMAX,5)=N
      IA(ISMAX,6)=IOHM
      IA(ISMAX,7)=0
      IA(ISMAX,8)=0
      IF(ITFLAG.NE.0)GOTO490
      IA(ISMAX,7)=ITTY
      IA(ISMAX,8)=IPHS
  490 IA(ISMAX,9)=0
      IA(ISMAX,10)=0
      IA(ISMAX,11)=50
      IA(ISMAX,12)=MLENG
      IFLAG=0
      IF(MLENG.EQ.I1)GOTO700
      IA(ISMAX,2)=ISMAX+1
      ISMAX=ISMAX+1
```

```
      GOTO4RC
  700 CONTINUE
*       NOW HAVE FINISHED LATERALS, PROCEED TO WRITE OUTFILE
  800 CONTINUE
*       ARRIVE HERE WHEN FINISHED CONSTRUCTING IA ARRAY WITH
*       ISMAX THE LAST OF THE CONTIGUOUSLY ASSIGNED SECTION
*       NUMBERS.  NOW WRITE TO OUTFILE
      WRITE(LU,11)1000,NID
      DO 810 I=1,ISMAX
  810 WRITE(LU,11)1000+I,I,(IA(I,J),J=1,12)
      PRINT,'%PGTE FILE ',OUFILE
      CALL DETACH(LU,ISTAT,)
  900 PRINT,'TYPE 1 TO CONTINUE OR CR TO STOP'
      READ,I
      IF(I.NE.0)GOTO200
      STOP
* I1=ILENL(N)
* I2=IDLEL(N)
* I3=IDISL(N)
* IDIM    DIMENSION LIMIT ON ARRAY IA, ISECT LE IDIM
* IDISL   DISTANCE FROM SOURCE TOLATERAL BRANCH - A VECTOR
* IDLEL   LENGTH OF SECTION ON LATERAL - AVECTOR
* IDLEN   LENGTH OF SECTION ON MAIN ROUTE
* ILENL   LENGTH OF LATERAL - A VECTOR
* ILENM   LENGH OF MAIN ROUTE - INITIAL SPECIFICATION
* ILNEW   LENGTH OF MAIN ROUTE - AUGMENTED BY POSSIBLE LATERAL
* ISECT   CURRENT SECTION NUMBER
* ISMAX   LAST SECTION NUMBER ASSIGNED
* ISMAM   LAST SECTION NUMBER ALONG MAIN ROUTE
* ITTY    DISTRIBUTION TRANSFORMER TYPE
* LENGT   VECTOR CONTINUING ACCUMULATED LENGTH TO SECTION (=INDEX POSITIO
* LTYPL   LINE TYPE ON MAIN ROUTE
* LTYPL   LINE TYPE ON LATERAL-A VECTOR
* LTYPM   LINE TYPE ON MAIN ROUTE
* MLENG   WORKING CURRENT TOTAL LENGTH VARIABLE
* NID     NETWORK IDENTIFICATION NUMBER
* NLAT    NUMBER OF LATERALS
      END
```

```
0010*# RUTH *=(ULIB,CORE=30K)LIBRARY3
0020  * THIS PROGRAM IS USED TO CREATE A REVERSE NETWORK
0030  *  THERE IS NO CONSISTANCY CHECKING PROVIDED
0040  *    THE ORIGINAL NETWORK IS ASSUMED TO BE CHECKED
0050  * IN A PROGRAM SUCH AS NTWKER
0060  * THE REVERSE PATH FOLLOWS J.T.GAJJAR'S ALGORITHM
0070  * DATED 2/18/80     UPTO 10 REVERSE NETWORKS CAN
0080  * BE CREATED FROM ONE ORIGINAL NETWORK
0090 1001 FORMAT(/"THIS PROGRAM CREATES REVERSE PATH NETWORKS"/
0100     & "ENTER NETWORK NUMBER - 2 DIGITS ")
0110 1002 FORMAT(/"NETWORK IDENTIFICATION NUMBER IS ",I5/)
0120 1003 FORMAT(" ENTER 1 IF WRONG, ELSE CR")
0130  111 FORMAT(6HCNWKIN,I2,1H;)
0140 1000 FORMAT(V)
0150 1005 FORMAT(/"NETWORK ID NUMBER IN ERROR"/
0160     &"SHOULD BE ",I3," BUT IS ",I3/"PROGRAM ABORTS")
0170 1006 FORMAT("JNWKIN DATA BASE READING ERROR ENCOUNTERED"/
0180     & "RECORD NUMBER ",I5," HAS LINE NUMBER ",I5/"PROGRAM ABORTS")
0190  1010 FORMAT(/"ENTER NEW ROOT NODE "/)
0200 1011 FORMAT(/"SOURCE FOR REVERSE NETWORK = ",I5/)
0210  2000 FORMAT(10X,5I5)
0220  1025  FORMAT(/"ENTER 5 LOAD NUMBER AT ROOT -ITTY,IPHSE,I3ECD,IPRIM,IZNGR
0230  1015  FORMAT(/"THERE IS ONLY ONE SECTION TO THE ROOT"/
0240       & "OF THE ORIGINAL NETWORK. REVERSAL SHOULD BE"/
0250       & "PERFORMED MANUALLY"//"FURTHER PROCESSING SUSPENDED")
0260  126   FORMAT(5HJNOUT,I2,2H0;)
0270   127  FORMAT(5HJNOUT,I2,I1,1H;)
0280   1030 FORMAT(I4,5I5,1X,7I5)
0290  1020  FORMAT(/"DO YOU WANT ANOTHER REVERSAL OF THE SAME NETWORK WITH "/
0300       &"A DIFFERENT SOURCE POINT ?"/"IF SO ENTER 1 ELSE CR"/)
0310  1021  FORMAT("YOU HAVE ALREADY CREATED 10 NETWORKS, THE MAXIMUM"/
0320       &" ALLOWED FROM ONE INPUT NETWORK"/"PROGRAM ABORTS")
0330  1022  FORMAT("DO YOU WANT REVERSAL FROM ANOTHER NETWORK ?"/
0340       &" IF SO ENTER 1 ELSE CR"/)
0350      DIMENSION IDESC(1800,12),ISTAK(1800),ILIST(1800)
0360      CHARACTER *9 MNTIN,MNTOUT
0370      DATA LDAIN/10/LDAOUT/11/MAXND/1800/KOUT/0/
0380    1 WRITE(6,1001)
0390      READ, NNUM
0400      WRITE(6,1002)NNUM
0410      WRITE(6,1003)
0420      READ, ITEST
0430      IF(ITEST.NE.0)GOTO1
0440 *
0450 *     BASE JNWKIN12(12 ARE LOCATIONS FOR NETWORK IDENTIFICATION
0460 *     NUMBER). THIS NETWORK IS READ FROM FILE
0470 *
0480  110 ENCODE(MNTIN,111)NNUM
0490      CALL OPENF(LDAIN,MNTIN,ISTAT,1,0,1)
```

```
0500       READ(LDAIN,1000)NLINE,KNUM
0510       IF(NNUM.EQ.KNUM)GO TO 113
0520       WRITE(6,1005)NNUM,KNUM
0530       STOP 1005
0540   113 DO 114 I=1,MAXND
0550        ISTAK(I) = 0
0560        ILIST(I) =0
0570        IDESC(I,1)=-10
0580   114 CONTINUE
0590        JROOT = 0
0600         I=1000
0610   115   I=I+1
0620        READ(LDAIN,1000,END=125)NLINE,ISECT,IPAR,ILSON,IRSON,
0630        & LTYP,ILENCD,IRES,ITTY,IPHSE,ISECD,IPRIM,IZNGR
0640        IF(NLINE.EQ.I)GOTO116
0650    *  WRITE(6,1006)I,NLINE
0660       STOP 1006
0670   116 IDESC(ISECT,1)=ISECT
0680       IDESC(ISECT,2) =IPAR
0690       IDESC(ISECT,3)=ILSON
0700       IDESC(ISECT,4)=IRSON
0710       IDESC(ISECT,5)=LTYP
0720         IDESC(ISECT,6)=ILENCD
0730         IDESC(ISECT,7)=IRES
0740         IDESC(ISECT,8)=ITTY
0750         IDESC(ISECT,9)=IPHSE
0760         IDESC(ISECT,10)=ISECD
0770         IDESC(ISECT,11)=IPRIM
0780         IDESC(ISECT,12)=IZNGR
0790       GO TO 115
0800  *
0810  *    NOW THE NETWORK HAS BEEN READ IN
0820  *    NOTE THERE IS NO CONSISTANCY CHECKING AT ALL
0830  *    THE ORIGINAL NETWORK HAS A ROOT NODE IDENTIFIED
0840  *    BY A 0 PARENT
0850  *    NEXT WE READ IN SOURCE NODE FOR REVERSE NETWORK
0860  *
0870   125 WRITE(6,1010)
0880       READ,NSORCE
0890       WRITE(6,1011)NSORCE
0900       WRITE(6,1003)
0910       READ, ITEST
0920       IF(ITEST.NE.0)GO TO 125
0930  *    NEXT WE CHECK TO SEE THAT THIS IS A VALID SOURCE NODE
0940  *    NEXT WE TRAVERSE THE NETWORK & MAKE IT DISJOINT AT
0950  *    AT THE NEW SOURCE NODE BY SETTING IDESC(ND,1)
0960  *    EQUAL TO -10 FOR ALL NODES BELOW NSORCE
0970  *    THIS EFFECTIVELY REMOVES THEM FROM THE NETWORK
0980  *    DESCRIPTION AND FROM THE OUTPUT FILE
0990       DO 140 I=1,MAXND
1000    *  IF(I.LT.15)WRITE(6,2000)(IDESC(I,J),J=1,5)
1010       IF(IDESC(I,1).LT.0)GO TO 140
```

```
1020        ND=IDESC(I,1)
1030   131  IF(IDESC(ND,2).EQ.NSORCE)GO TO 132
1040        IF(IDESC(ND,2).EQ.0)GO TO 140
1050        ND = IDESC( D,2)
1060        GO TO 131
1070   132  IDESC(I,1)=-10
1080   140  CONTINUE
1090        I=1
1100        NS=NSORCE
1110   141  ISTAK(I)=NS
1120   *    PRINT,I,ISTAK(I),NS,IDESC(NS,2)
1130        IF(IDESC(NS,2).EQ.0)GO TO 200
1140        NPAR=IDESC(NS,2)
1150        IF(IDESC(NPAR,3).EQ.NS)GO TO 142
1160        ILIST(I)=IDESC(NPAR,3)
1170        GO TO 144
1180   142  ILIST(I) =IDESC(NPAR,4)
1190   144  IF(ILIST(I).EQ.0)GO TO 145
1200        IDESC(ILIST(I),2)=NS
1210   145  NS=IDESC(NS,2)
1220        I=I+1
1230        GO TO 141
1240   200  NLE=I
1250        IF(NLE.LE.1)GO TO 300
1260        IDESC(NSORCE,2)=0
1270        IDESC(NSORCE,3)=ISTAK(2)
1280        IDESC(NSORCE,4)=ILIST(1)
1290          DO 201 K=8,12
1300        IDESC(NSORCE,K)=IDESC(ISTAK(2),K)
1310    201   CONTINUE
1320            IF(NLE.EQ.2)GO TO 230
1330        DO 220 I=2,NLE-1
1340        NS=ISTAK(I)
1350        IDESC(NS,2)=ISTAK(I-1)
1360        IDESC(NS,3)=ISTAK(I+1)
1370        IDESC(NS,4)=ILIST(I)
1380           DO 219 K=8,12
1390        IDESC(NS,K)=IDESC(ISTAK(I+1),K)
1400    219   CONTINUE
1410   220   CONTINUE
1420           IDESC(ISTAK(NLE),2) =ISTAK(NLE-1)
1430     222  IDESC(ISTAK(NLE),3) = 0
1440           IDESC(ISTAK(NLE),4) = 0
1450           WRITE(6,1025)
1460           READ,(IDESC(ISTAK(NLE),K),K=8,12)
1470   *     NEXT WE CREATE THE OUTPUT FILE AND WRITE THE NEW
1480   *     NETWORK DESCRIPTION DOWN INTO OUT PUT FILE
1490   *       BEFORE THAT WE TAKE CARE OF THE SITUATION VWHERE THERE
1500   *       ARE TWO OR LESS SECTIONS TO THE ORIGINAL SOURCE
1510   *       IN THAT CASE THE ALGORITHMIC APPROACH USED HERE WILL
1520   *      FAIL IN THE FORTRAN LOGIC. IT IS EASY TO TAKE CARE OF
1530   *       THIS SITUATION MANUALLY. WE WILL SIMPLY ABORT WITH
```

```
1540 *       AN APPROPRIATE ERROR MESSAGE.
1550         GO TO 250
1560    230 IDESC(ISTAK(NLE),2)=NSORCE
1570         GO TO 222
1580    300  WRITE(6,1015)
1590         GO TO 261
1600 *       NOW THE OUTPUT FILE IS CREATED WITH ITS HEADER
1610 *       THE NAME OF THE OUTPUT NETWORK FILE IS JNOUT123
1620 *       WHERE 12 ARE THE IDENTIFICATION NUMBERS OF THE
1630 *       THE INPUT NETWORK AND 3 IS THE ONE DIGIT OUTPUT
1640 *       NETWORK CODE  - KOUT, STARTING WITH 0
1650    250 IF(KOUT.EQ.0)ENCODE(MNTOUT,126)NNUM
1660         IF(KOUT.NE.0)ENCODE(MNTOUT,127)NNUM,KOUT
1670         CALL OPENF(LDAOUT,MNTOUT,ISTAT)
1680         IF(ISTAT.NE.0)CALL FILSTO(MNTOUT,'STATUS',0,ISTAT,250)
1690        PRINT,'OUTPUT FILENAME=',MNTOUT
1700         NLINE=1000
1710         WRITE(LDAOUT,1030)NLINE,NNUM,KOUT,NSORCE
1720         DO 260 I=1,MAXND
1730         IF(IDESC(I,1).LT.0)GO TO 260
1740         NLINE =NLINE+1
1750        IF(NLINE.GT.9999)STOP 9999
1760         WRITE(LDAOUT,1030)NLINE,(IDESC(I,J),J=1,12)
1770    260 CONTINUE
1780         CALL DETACH(LDAOUT,ISTAT,)
1790         IF(ISTAT.NE.0)CALL FILSTO(MNTOUT,'DETACH',0,ISTAT,260)
1800    261  WRITE(6,1020)
1810    -    READ, ITEST
1820         IF(ITEST.EQ.0)GO TO 270
1830         KOUT = KOUT+1
1840         IF(KOUT.GE.10)WRITE(6,1021)
1850         IF(KOUT.GE.10)STOP 270
1860         REWIND LDAIN
1870         GO TO 110
1880    270 WRITE(6,1022)
1890          READ, ITEST
1900         IF(ITEST.EQ.0)STOP 280
1910         CALL DETACH(LDAIN,ISTAT,)
1920         IF(ISTAT.NE.0)CALL FILSTO(MNTIN,'STATUS',0,ISTAT,280)
1930         GO TO 1
1940         END
```

```
0010 *        THIS PROGRAM SERVES TO CONSOLIDATE SEPERATE NEUTRAL
0020 *        CONDUCTORS ASSOCIATED WITH INDIVIDUAL UNDERGROUND
0030 *        POWER CABLES INTO ELECTRICALLY EQUIVALENT SINGLE NEUTRAL
0040 *        CONDUCTOR ANALOGOUS TO THE SITUATION FOUND IN OPEN WIRE
0050 *        OVERHEAD CONDUCTORS.THIS PROGRAM USES IMPEDANCE AND
0060 *        ADMITTANCE DATA IN UNIT OF OHMS/METER AND MHOS/METER
0070 *        CALCULATED FOR BURIED CABLE BY USE OF THE BPA EMTP
0080 *        CABLE CONSTANTS OPTION.
0090 *        IN CALCULATION PERFORMED BY THE PROGRAM THE ELECTRICAL
0100 *        EQUIVALENT NEUTRAL CONDUCTOR IS NOT ASSUMED TO BE AT
0110 *        GROUND POTENTIAL THUS RETAINING THE EVUIVALENT NEUTRAL
0120 *        AS AN 'EXPLICIT' CONDUCTOR.
0130 *          THIS PROGRAM IS EQUIPPED TO SUPPLY THE
0140 *        PROPER Z AND Y MATRICES FOR EITHER 1,2 OR 3 PHASE CASE BY
0150 *        THE PROPER EXTRACTION AND MANIPULATION OF THE ELEMENTS OF
0160 *        THE 6 BY 6 IMPEDANCE AND ADMITTANCE MATRICES SUPPLIED BY
0170 *        THE B.P.A. PROGRAM
0180 *        WRITTEN BY RICHARD WOODING
0190     10 FORMAT(V)
0200     20 FORMAT(I4,10I5)
0210     30 FORMAT(I4,1X,1PE15.8,1X,E15.8)
0220     40 FORMAT(I4)
0230        NFN=9999
0240        DATA LUIN/11/,LUOU/12/
0250        CHARACTER*9 ROW,IMPED,ADMIT
0260        CHARACTER*9 INFILE,OUTFILE
0270        COMPLEX CC,CM1,CM2,CM3,CM4,ZMOD5,ZMOD6
0280 *        DATAZ IS THE IMPEDANCE MATRIX,DATAY IS THE ADMITTANCE MATRIX
0290        DIMENSION DATA(24,6),DATAZ(6,6),DATAY(6,6),ZMOD(4),ZMOD6(4),
0291       &          DATZ2(6,6),DATY2(6,6),DATZ3(6,6)
0300        COMPLEX DATAZ,DATAY,ZMOD,C1,C2,C3,DATZ2,DATY2,DATZ3
0310    110 PRINT,'ENTER THE NAME OF THE COMPLEX DATA FILE FOLLOWED BY;'
0320        READ,INFILE
0330        CALL OPENF(LUIN,INFILE,ISTAT,3,0,1)
0340        PRINT,'ENTER THE NAME OF THE OUTPUT FILE FOLLOWED BY A ;'
0350        READ,OUTFILE
0360        CALL OPENF(LUOU,OUTFILE,ISTAT,3,0,0)
0370     READ(LUIN,10,END=999)NPR1,NOCASE,NPC,NFOC,NCC,NTPC,NTOT,NPHSET,NSP,NSN
0380       &       ,NEUF
0390        READ(LUIN,10)LAUNO,IMPED
0400        PRINT,'LTYPE CASE DATA'
0410        PRINT,NPR1,NOCASE,NPC,NFOC,NCC,NTPC,NTOT,NPHSET,NSP,NSN,NEUF
0420        PRINT,'TYPE A CR IF OK'
0430        READ,NCHK
0440        IF(NCHK.EQ.0)GOTO99
0450        PRINT,'ENTER MODIFIED DATA CASE'
0460        READ,NPR1,NOCASE,NPC,NFOC,NCC,NTPC,NTOT,NPHSET,NSP,NSN,NEUF
0470        PRINT,'MODIFIED CASE DATA'
0480        PRINT,NPR1,NOCASE,NPC,NFOC,NCC,NTPC,NTOT,NPHSET,NSP,NSN,NEUF
```

```
0490        PRINT,'TYPE CR IF O.K.'
0500        READ,NCHK
0510        IF(NCHK.EQ.0)GOTO99
0520        IF(NCHK.NE.0)GOTO280
0530     99 LINX=1
0540        DO 100 I=1,11,2
0550        READ(LUIN,10)LNUM1,ROW,(DATA(I,J),J=1,6)
0560        READ(LUIN,10)LNUM2,(DATA(I+1,J),J=1,6)
0570        READ(LUIN,10)LNUM3,LONE
0580        READ(LUIN,10)LNUM4
0590   0100 CONTINUE
0600        READ(LUIN,10) LNUM0,ADMIT
0610        DO 101 I=13,23,2
0620        READ(LUIN,10)LNUM1,ROW,(DATA(I,J),J=1,6)
0630        READ(LUIN,10)LNUM2,(DATA(I+1,J),J=1,6)
0640        READ(LUIN,10)LNUM3,LONE
0650        READ(LUIN,10)LNUM4
0660     101 CONTINUE
0670        PRINT,'TYPE A 1 TO BYPASS THIS CASE'
0680        READ,NPASS
0690        IF(NPASS.EQ.1)GOTO110
0700        PRINT,'ENTER THE NUMBER OF PHASE CABLES'
0710        READ,NPHASE
0720        NOC=NPHASE*2
0730     111 NPR1=1010
0740  *       HERE THE COMPLEX 6X6 MATRIX IS READ FROM THE DATA FILE
0750        DO 120 I=1,6
0760        DO 130 J=1,6
0770        IP=(2*I)-1
0780        IQ=(2*I)
0790        RC1=DATA(IP,J)
0800        RC2=DATA(IQ,J)
0810        DATAZ(I,J)=CMPLX(RC1,RC2)
0820     130 CONTINUE
0830     120 CONTINUE
0840  *       NOW FORM THE COMPLEX 6X6 Y MATRIX
0850        DO 140 I=1,6
0860        DO 150 J=1,6
0870        KP=(((2*I)-1)+12)
0880        KQ=((2*I)+12)
0890        DATAY(I,J)=CMPLX(DATA(KP,J),DATA(KQ,J))
0900     150 CONTINUE
0910     140 CONTINUE
0920        IF(NPHASE.EQ.2)GOTO205
0930        IF(NPHASE.EQ.1)GOTO270
0940  *       NOW FORM THE COEFFICENT COMMON TO ALL THE MODIFIERS
0950        Q1=(DATAZ(5,6)+DATAZ(4,4)-DATAZ(4,5)-DATAZ(4,6))
0960        Q2=((2*DATAZ(4,4))-(2*DATAZ(4,5)))
0970        Q3=((2*DATAZ(4,4))-(2*DATAZ(4,6)))
0980        CC=((Q2*Q3)-(Q1)**2)
0990  *       NOW COMMON ELEMENTS IN EACH MODIFIER ARE FORMED
1000  *       NOW FORM THE MODIFIERS
```

```
1010        DO 160 I=1,4
1020        DO 170 J=1,4
1030        CM1=(((DATAZ(I,5)-DATAZ(I,4))/CC)*(((DATAZ(J,6)-DAT
1040       &        AZ(J,4))*01)-((DATAZ(J,5)-DATAZ(J,4))*03))
1050        CM2=(((DATAZ(I,6)-DATAZ(I,4))/CC)*(((DATAZ(J,5)-DAT
1060       &        AZ(J,4))*01)-((DATAZ(J,6)-DATAZ(J,4))*02))
1070        DATZ3(I,J)=DATAZ(I,J)+(CM1+CM2)
1080    170 CONTINUE
1090    160 CONTINUE
1100 *        NOW FORM THE MODIFIED Y MATRIX
1110        DO 165 I=1,3
1120        DO 166 J=1,3
1130        DATY2(I,J)=DATAY(I,J)
1140    166 CONTINUE
1150    165 CONTINUE
1160        DO 190 I=1,3
1170        DATY2(I,4)=DATAY(I,4)+DATAY(I,5)+DATAY(I,6)
1180    190 CONTINUE
1190        DO 200 K=1,3
1200        DATY2(4,K)=DATAY(K,4)+DATAY(K,5)+DATAY(K,6)
1210    200 CONTINUE
1220        DATY2(4,4)=(DATAY(4,4)+(2*DATAY(4,5))+(2*DATAY(4,6))+DATAY(5,5)
1230       &        +(2*DATAY(5,6))+DATAY(6,6))
1240    205 NPM=NPHASE+1
1250        DO 206 I=1,2
1260        DATZ2(I,3)=DATAZ(I,4)
1270        DATZ2(3,I)=DATAZ(4,I)
1280    206 CONTINUE
1290        DATZ2(3,4)=DATAZ(4,5)
1300        DO 207 J=1,2
1310        DATZ2(J,4)=DATAZ(J,5)
1320        DATZ2(4,J)=DATAZ(5,J)
1330    207 CONTINUE
1340        DATZ2(4,3)=DATAZ(5,4)
1350        DATZ2(3,3)=DATAZ(4,4)
1360        DATZ2(4,4)=DATAZ(5,5)
1370        DO 208 I=1,2
1380        DATZ2(I,I)=DATAZ(I,I)
1390    208 CONTINUE
1400        DATZ2(1,2)=DATAZ(1,2)
1410        DATZ2(2,1)=DATAZ(2,1)
1420        IF(NPHASE.NE.2)GOTO270
1430        DO 215 I=1,NPM
1440        DO 216 J=1,NPM
1450        CM3=((DATZ2(I,4)-DATZ2(I,3))*(DATZ2(J,3)-DATZ2(J,4)))
1460        CM4=(((2*DATZ2(3,4))-(DATZ2(3,3))-(DATZ2(4,4))))
1470        ZMOD5=-(CM3/CM4)
1480        DATZ3(I,J)=DATZ2(I,J)+ZMOD5
1490    216 CONTINUE
1500    215 CONTINUE
1510        DATY2(1,1)=DATAY(1,1)
1520        DATY2(1,2)=DATAY(1,2)
```

```
      DATY2(2,1)=DATAY(2,1)
      DATY2(2,2)=DATAY(2,2)
      DATY2(1,3)=DATAY(1,4)+DATAY(1,5)
      DATY2(2,3)=DATAY(2,4)+DATAY(2,5)
      DO 221 I=1,2
      DATY2(3,I)=DATAY(4,I)+DATAY(5,I)
  221 CONTINUE
      DATY2(3,3)=(DATAY(4,4)+DATAY(5,5)+(2*DATAY(4,5)))
*       NOW WRITE THE OUTPUT FILE
  270 WRITE(LUOU,20)NPR1,NOCASE,NPC,NFC,NCC,NTPC,NTOT,NPHSET,NSP
     &      ,NSI,NEOF
      IF(NPHASE.EQ.2)GOTO400
      NP1=NPHASE+1
      IF(NPHASE.EQ.1)DATZ3(1,1)=DATAZ(1,1)
      IF(NPHASE.EQ.1)DATZ3(2,1)=DATAZ(4,1)
      IF(NPHASE.EQ.1)DATZ3(1,2)=DATAZ(1,4)
      IF(NPHASE.EQ.1)DATZ3(2,2)=DATAZ(4,4)
      IF(NPHASE.EQ.1)DATY2(1,1)=DATAY(1,1)
      IF(NPHASE.EQ.1)DATY2(1,2)=DATAY(1,4)
      IF(NPHASE.EQ.1)DATY2(2,1)=DATAY(4,1)
      IF(NPHASE.EQ.1)DATY2(2,2)=DATAY(4,4)
      DO 210 K=1,NPM
      DO 220 L=1,NPM
      NPR1=NPR1+10
      WRITE(LUOU,30)NPR1,DATZ3(K,L)
  220 CONTINUE
  210 CONTINUE
  400 NS=1
      IF(NPHASE.NE.2)GOTO430
      DO 410I=1,NPM
      DO 420 J=1,NPM
      NPR1=NPR1+10
      WRITE(LUOU,30)NPR1,DATZ3(I,J)
  420 CONTINUE
  410 CONTINUE
  430 NT=1
      DO 230 K=1,NPM
      DO 240 L=1,NPM
      NPR1=NPR1+10
      WRITE(LUOU,30)NPR1,DATY2(K,L)
  240 CONTINUE
  230 CONTINUE
      WRITE(LUOU,40)NFN
      PRINT,'TYPE A 1 TO PROCESS A NEW INPUT FILE'
      READ,I
      CALL DETACH(LUIN,ISTAT,)
      CALL DETACH(LUOU,ISTAT,)
      IF(I.EQ.1)GOTO110
  280 PRINT,'STOP OF EXECUTION NORMAL'
  999 PRINT,'END OF DATA FILE'
      STOP
      END
```

```
010**RUN**=;NEURED02(I.CGO,CORE=30K)
020       SUBROUTINE NEURED(Y,Z,L,M,NDIMZY,SA1,SA2,SA3,WKAREA,NW,NDIM)
030 *       THIS SUBROUTINE COMPUTES A NEUTRAL REDUCTION PROCEDURE
040 *       USING A GENERALIZED MATRIX APPROACH DEVELOPED BY R.RUSTAY
050 *       NOTE THAT THE RESULTS ARE RETURNED IN Z AND Y WITH ZEROED
060 *       BORDERS.  THUS ORIGINAL Z,Y, ARE DESTROYED
070       COMPLEX SA1(NDIM,NDIM),Y(NDIMZY,NDIMZY),SA3(NDIM,NDIM)
080       COMPLEX SA2(NDIM,NDIM),Z(NDIMZY,NDIMZY)
090       COMPLEX SUM,ZERO,ZONE,DEN
100       DIMENSION WKAREA(NW)
110       DATA ZERO/(0.,0.)/,ZONE/(1.,0.)/
120 *       IF M=N+1 THERE IS NO NEED FOR COMPUTATION
130       IF(M.EQ.(N+1))RETURN
140       L=N+1
150       K=M-N
160 *       REDUCTION OF Y
170       DO 210 I=1,N
180       SUM=ZERO
190       DO 200 J=L,M
200    200 SUM=SUM+Y(I,J)
210       Y(I,L)=SUM
220    210 Y(L,I)=SUM
230       SUM=ZERO
240       DO 220 I=L,M
250       DO 220 J=L,M
260    220 SUM=SUM+Y(I,J)
270       Y(L,L)=SUM
280 *       THIS NEXT STEP OF ZEROING BORDER IS NOT NECESSARY
290       CALL CMTZRO(Y,L,L,NDIMZY)
300 *       REDUCTION OF Z
310 *         EXTRACT Z22 AND Z12 AND PLACE IN SA1 AND SA2 RESPECTIVELY
320       DO 320 J=1,K
330       JPN=J+N
340       DO 300 I=1,K
350    300 SA1(I,J)=Z(I+N,JPN)
360       DO 310 I=1,N
370    310 SA2(I,J)=Z(I,JPN)
380    320 CONTINUE
390 *       NEXT INVERT Z22 USING SA1 AND COMPUTE DEN=ST*Z22I*S
400       CALL CMTINV(SA1,K,K,NDIM,WKAREA)
410       DEN=ZERO
420       DO 330 I=1,K
430       DO 330 J=1,K
440    330 DEN=SUM+SA1(I,K)
450 *         NEXT COMPUTE Z12*Z22I (NXM-N) AND SAVE IN SA3.  WILL THEN NO
460 *         LONGER NEED Z22I=SA1 SO THAT SA1 WILL BECOME SCRATCH
470       CALL CMTMPY(SA2,SA1,SA3,N,K,K,NDIM)
480 *         NOW SA3=Z12*Z22I, SA1 IS SCRATCH, SA2=Z12
490 *         NEXT COMPUTE SA1=Z12*Z22I*Z12T=SA3*Z12T=SA3*SA2T
```

```
      CALL CNTMPT(SA3,SA2,SA1,K,N,N,NDIM)
*          NOW SA1=Z12*Z22I*Z12T, SA3=Z12*Z22I, SA2 IS SCRATCH
*          NEXT COMPUTE Z11-Z12*Z22I*Z12T AND DO IN PLACE
      DO 350 I=1,N
      DO 350 J=1,N
  350 Z(I,J)=Z(I,J)-SA1(I,J)
*          NOW SA1 AND SA2 ARE SCRATCH.  REMEMBER SA3=Z12*Z22I
*          NEXT COMPUTE SA1=Z12*Z22I*S. IE SA1 BECOMES AN NX1 VECTOR
      DO 370 I=1,N
      SUM=ZERO
      DO 360 J=1,K
  360 SUM=SUM+SA3(I,J)
  370 SA1(I,1)=SUM
*          NOW SA1=Z12*Z22I*S AN N-VECTOR, SA2 AND SA3 ARE SCRATCH
*          NEXT COMPUTE SA2=Z12*Z22I*S*ST*Z22I*Z12T=SA1*SA1T
      CALL CNTMPT(SA1,SA1,SA2,1,N,N,NDIM)
*          NOW FORM Z11E, REMEMBERING Z11-Z12*Z22I*Z12T HAS ALREADY
*          BEEN DONE.  ALSO SA1=Z12*Z22I*S AN N-VECTOR,
*          SA2=Z12*Z22I*S*ST*Z22I*Z12T AND SA3 IS SCRATCH
      DO 420 I=1,N
      DO 400 J=1,N
  400 Z(I,J)=Z(I,J)*SA2(I,J)/DEN
*          NOW FILLING IN REMAINING PARTITIONS
      Z(I,L)=SA1(I,1)/DEN
  420 Z(L,I)=S(I,L)
      Z(L,L)=ZONE/DEN
*          ZEROING BORDER
      CALL CNTZRO(Z,L,L,NDIMZY)
      RETURN
      END
```

NOTE: This Program Has Not Been Debugged and is Not Operational.

```
0001*# RUN =NEUREDH3(ULIB,CORE=35K)LIBRARY3:LIBRARY/IMSL,R:LIBRARY/OLDTSLIB,R
1000 *       THIS PROGRAM SERVES TO CONSOLIDATE SEPERATE NEUTRAL
1010 *       CONDUCTORS ASSOCIATED WITH INDIVIDUAL UNDERGROUND
1020 *       POWER CABLES INTO ELECTRICALLY EQUIVALENT SINGLE NEUTRAL
1030 *       CONDUCTOR ANALOGOUS TO THE SITUATION FOUND IN OPEN WIRE
1040 *       OVERHEAD CONDUCTORS.THIS PROGRAM USES IMPEDANCE AND
1050 *       ADMITTANCE DATA IN UNIT OF OHMS/METER AND MHOS/METER
1060 *       CALCULATED FOR BURIED CABLE BY USE OF THE BPA EMTP
1070 *       CABLE CONSTANTS OPTION.
1080 *       IN CALCULATION PERFORMED BY THE PROGRAM THE ELECTRICAL
1090 *       EQUIVALENT NEUTRAL CONDUCTOR IS NOT ASSUMED TO BE AT
1100 *       GROUND POTENTIAL THUS RETAINING THE EQUIVALENT NEUTRAL
1110 *       AS AN 'EXPLICIT' CONDUCTOR.
1120 *         THIS PROGRAM IS EQUIPPED TO SUPPLY THE
1130 *       PROPER Z AND Y MATRICES FOR EITHER 1,2 OR 3 PHASE CASE BY
1140 *       THE PROPER EXTRACTION AND MANIPULATION OF THE ELEMENTS OF
1150 *       THE 6 BY 6 IMPEDANCE AND ADMITTANCE MATRICES SUPPLIED BY
1160 *       THE E.P.A. PROGRAM
1170 *       WRITTEN BY RICHARD WOODING
1180     10 FORMAT(V)
1190     20 FORMAT(I4,10I5)
1200     30 FORMAT(I4,1X,1PE15.8,1X,E15.8)
1210     40 FORMAT(I4)
1220        NFN=9999
1230        DATA LUIN/11/,LUOU/12/
1240        CHARACTER*9 ROW,IMPED,ADMIT
1250        CHARACTER*9 INFILE,OUTFILE
1260        COMPLEX CC,CM1,CM2,CM3,CM4,ZMOD5,ZMOD6
1270 *        DATAZ IS THE IMPEDANCE MATRIX,DATAY IS THE ADMITTANCE MATRIX
1280        DIMENSION DATA(24,6),DATAZ(6,6),DATAY(6,6),ZMOD(4),ZMOD6(4),
1281       &          DATZ2(6,6),DATY2(6,6),DATZ3(6,6)
1290        COMPLEX DATAZ,DATAY,ZMOD,Q1,Q2,Q3,DATZ2,DATY2,DATZ3
1300        DIMENSION DENT(7)
1310        PRINT,'IF THE MANUAL ENTRY MODE IS DESIRED ENTER A 1,IF NOT CR'
1320        READ,MPMODE
1325   110  CONTINUE
1330        IF(MPMODE.EQ.1)GO TO 97
1340        PRINT,'ENTER THE NAME OF THE COMPLEX DATA FILE FOLLOWED BY;'
1350        READ,INFILE
1360        CALL OPENF(LUIN,INFILE,ISTAT,3,0,1)
1370     97 PRINT,'ENTER THE NAME OF THE OUTPUT FILE FOLLOWED BY A ;'
1380        READ,OUTFILE
1390        CALL OPENF(LUOU,OUTFILE,ISTAT,3,0,0)
1395        IF(MPMODE.EQ.1) GO TO 98
1400     READ(LUIN,10,END=999)NPR1,NOCASE,NPC,NFGC,NCC,NTPC,NTOT,NPHSET,NSP,NSN
1410       &          NEUF
1420        READ(LUIN,10)LNUM0,IMPED
1430        PRINT,'LTYPE CASE DATA'
1440        PRINT,NPR1,NOCASE,NPC,NFGC,NCC,NTPC,NTOT,NPHSET,NSP,NSN,NEUF
```

```
      PRINT,'TYPE A CR IF OK'
      READ,ICHK
      IF(ICHK.EQ.0)GOTO99
   99 PRINT,'E TER MODIFIED DATA CASE'
      READ,IPR1,NOCASE,NRC,NPC,NCC,NTPC,NTOT,NPHSET,NSP,NSN,NEUF
      PRINT,'MODIFIED CASE DATA'
      PRINT,IPR1,NOCASE,NRC,NPC,NCC,NTPC,NTOT,NPHSET,NSP,NSN,NEUF
      PRINT,'TYPE CR IF O.K.'
      READ,ICHK
      IF(ICHK.EQ.0)GOTO99
      IF(ICHK.NE.0)GOTO240
   98 CONTINUE
      IF(NPMODE.EQ.1) GO TO 500
      LINX=1
      DO 100 I=1,11,2
      READ(LUIN,10)LNUM1,ROW,(DATA(I,J),J=1,6)
      READ(LUIN,10)LNUM2,(DATA(I+1,J),J=1,6)
      READ(LUIN,10)LNUM3,LONE
      READ(LUIN,10)LNUM4
  100 CONTINUE
      READ(LUIN,10) LNUMO,ADMIT
      DO 101 I=13,23,2
      READ(LUIN,10)LNUM1,ROW,(DATA(I,J),J=1,6)
      READ(LUIN,10)LNUM2,(DATA(I+1,J),J=1,6)
      READ(LUIN,10)LNUM3,LONE
      READ(LUIN,10)LNUM4
  101 CONTINUE
  500 CONTINUE
      IF(NPMODE.NE.1) GO TO 102
      PRINT,'ENTER RCC'
      READ,DENT(1)
      PRINT,'ENTER RNN'
      READ,DENT(2)
      PRINT,'ENTER XCC'
      READ,DENT(3)
      PRINT,'ENTER XNN'
      READ,DENT(4)
      PRINT,'ENTER RAB'
      READ,DENT(5)
      PRINT,'ENTER XAB'
      READ,DENT(6)
      PRINT,'ENTER XCN'
      READ,DENT(7)
      PRINT,'ENTER YCC'
      READ,DENT(8)
      PRINT,'ENTER YNN'
      READ,DENT(9)
      J=1
      K1=1
      K2=2
      K3=3
      DO 41 I=1,11,2
```

```
1960          K1=I+1
1970          DATA(I,J)=DENT(K1)
1980          DATA(K1,J)=DENT(K3)
1990          J=J+1
2000          IF(I.EQ.5) K1=K1+1
2010          IF(I.EQ.5) K3=K3+1
2020       41 CONTINUE
2025          J=2
2030          DO 43 I=1,9,2
2050          DATA(I,J)=DENT(5)
2060          IA=I+1
2070          DATA(IA,J)=DENT(6)
2080          IB=I+2
2090          JB=J-1
2100          DATA(IB,JB)=DENT(5)
2110          IC=IB+1
2120          DATA(IC,JB)=DENT(6)
2130          J=J+1
2140       43 CONTINUE
2145          J=3
2150          DO 45 I=1,7,2
2170          DATA(I,J)=DENT(5)
2180          IE=I+1
2185          DATA(IE,J)=DENT(6)
2190          JE=J-2
2195          IZ=I+4
2200          DATA(IZ,JE)=DENT(5)
2210          IF=IZ+1
2220          DATA(IF,JE)=DENT(6)
2230          J=J+1
2240       45 CONTINUE
2245          J=4
2250          DO 47 I=1,5,2
2270          DATA(I,J)=DENT(5)
2280          IE=I+1
2290          DATA(IE,J)=DENT(7)
2300          IG=I+6
2310          JF=J-3
2320          DATA(IG,JF)=DENT(5)
2330          IH=IG+1
2340          DATA(IH,JF)=DENT(7)
2350          J=J+1
2360       47 CONTINUE
2365          J=5
2370          DO 49 I=1,3,2
2390          DATA(I,J)=DENT(5)
2400          IP=I+1
2410          DATA(IP,J)=DENT(6)
2420          IQ=I+8
2430          JQ=J-4
2440          DATA(IQ,JQ)=DENT(5)
2450          IR=IQ+1
```

```
2460        DATA(IR,IC)=DENT(6)
2470        J=J+1
2480     49 CONTINUE
2490        DATA(1,6)=DENT(5)
2500        DATA(2,6)=DENT(6)
2510        DATA(11,1)=DENT(5)
2520        DATA(12,1)=DENT(6)
2530        DO 51 I=13,24
2540        DO 52 J=1,6
2550        DATA(I,J)=0.
2560     52 CONTINUE
2570     51 CONTINUE
2575        J=1
2580        DO 53 I=14,18,2
2600        DATA(I,J)=DATA(I,J)+DENT(8)
2610        J=J+1
2620     53 CONTINUE
2630        MXD=4
2640        DO 55 I=14,18,2
2650        DATA(I,MXD)=DATA(I,MXD)+DENT(8)
2660        MXD=MXD+1
2670     55 CONTINUE
2680        MXD=1
2690        DO 56 I=20,24,2
2700        DATA(I,MXD)=DATA(I,MXD)+DENT(8)
2710       MXD=MXD+1
2720     56 CONTINUE
2730        MXD=4
2740        DO 57 I=20,24,2
2750        DATA(I,MXD)=DATA(I,MXD)+DENT(9)
2760        MXD=MXD+1
2770     57 CONTINUE
2780    102 PRINT,'TYPE A 1 TO BYPASS THIS CASE'
2790        READ,IPASS
2800        IF(IPASS.EQ.1)GOTO110
2810        PRINT,'ENTER THE NUMBER OF PHASE CABLES'
2820        READ,NPHASE
2830        NOC=NPHASE*2
2840    111 NPR1=1010
2850 *        HERE THE COMPLEX 6X6 MATRIX IS READ FROM THE DATA FILE
2860        DO 120 I=1,6
2870        DO 130 J=1,6
2880        IP=(2*I)-1
2890        IQ=(2*I)
2900        RC1=DATA(IP,J)
2910        RC2=DATA(IQ,J)
2920        DATAZ(I,J)=CMPLX(RC1,RC2)
2930    130 CONTINUE
2940    120 CONTINUE
2950 *        NOW FORM THE COMPLEX 6X6 Y MATRIX
2960        DO 140 I=1,6
2970        DO 150 J=1,6
```

```
2980        KP=(((2*I)-1)*12)
2990        KQ=((2*I)*12)
3000        DATAY(I,J)=CMPLX(DATA(KP,J),DATA(KQ,J))
3010    150 CONTINUE
3020    140 CONTINUE
9999        CALL NEURED(DATAY,DATAZ,4,5,6,SA1,SA2,SA3,MKAREA,20,6)
9999        DO 141 I=1,6
9999        DO 142 J=1,6
9999        PRINT,DATAZ(I,J)
9999        PRINT,DATAY(I,J)
9999    142 CONTINUE
9999    141 CONTINUE
3030        IF(NPHASE.EQ.2)GOTO205
3040        IF(NPHASE.EQ.1)GOTO270
3050 *        NOW FORM THE COEFFICENT COMMON TO ALL THE MODIFIERS
3060        Q1=(DATAZ(5,6)+DATAZ(4,4)-DATAZ(4,5)-DATAZ(4,6))
3070        Q2=((2*DATAZ(4,4))-(2*DATAZ(4,5)))
3080        Q3=((2*DATAZ(4,4))-(2*DATAZ(4,6)))
3090        CC=((Q2*Q3)-(Q1)**2)
3100 *        NOW COMMON ELEMENTS IN EACH MODIFIER ARE FORMED
3110 *        NOW FORM THE MODIFIERS
3120        DO 160 I=1,4
3130        DO 170 J=1,4
3140        CM1=(((DATAZ(I,5)-DATAZ(I,4))/CC)*(((DATAZ(J,6)-DAT
3150       &        AZ(J,4))*Q1)-(DATAZ(J,5)-DATAZ(J,4))*Q3))
3160        CM2=(((DATAZ(I,6)-DATAZ(I,4))/CC)*(((DATAZ(J,5)-DAT
3170       &        AZ(J,4))*Q1)-(DATAZ(J,6)-DATAZ(J,4))*Q2))
3180        DATZ3(I,J)=DATAZ(I,J)+(CM1+CM2)
3190    170 CONTINUE
3200    160 CONTINUE
3210 *        NOW FORM THE MODIFIED Y MATRIX
3220        DO 165 I=1,3
3230        DO 166 J=1,3
3240        DATY2(I,J)=DATAY(I,J)
3250    166 CONTINUE
3260    165 CONTINUE
3270        DO 190 I=1,3
3280        DATY2(I,4)=DATAY(I,4)+DATAY(I,5)+DATAY(I,6)
3290    190 CONTINUE
3300        DO 200 K=1,3
3310        DATY2(4,K)=DATAY(K,4)+DATAY(K,5)+DATAY(K,6)
3320    200 CONTINUE
3330        DATY2(4,4)=(DATAY(4,4)+(2*DATAY(4,5))+(2*DATAY(4,6))+DATAY(5,5)
3340       &       +(2*DATAY(5,6))+DATAY(6,6))
3350    205 NPM=NPHASE+1
3360        DO 206 I=1,2
3370        DATZ2(I,3)=DATAZ(I,4)
3380        DATZ2(3,I)=DATAZ(4,I)
3390    206 CONTINUE
3400        DATZ2(3,4)=DATAZ(4,5)
3410        DO 207 J=1,2
3420        DATZ2(J,4)=DATAZ(J,5)
```

```
3430        DATZ2(4,J)=DATAZ(5,J)
3440    207 CONTINUE
3450        DATZ2(4,3)=DATAZ(5,4)
3460        DATZ2(3,3)=DATAZ(4,4)
3470        DATZ2(4,4)=DATAZ(5,5)
3480        DO 208 I=1,2
3490        DATZ2(I,I)=DATAZ(I,I)
3500    208 CONTINUE
3510        DATZ2(1,2)=DATAZ(1,2)
3520        DATZ2(2,1)=DATAZ(2,1)
3530        IF(NPHASE.NE.2)GOTO270
3540        DO 215 I=1,NPM
3550        DO 216 J=1,NPM
3560        CM3=((DATZ2(I,4)-DATZ2(I,3))*(DATZ2(J,3)-DATZ2(J,4)))
3570        CM4=(((2*DATZ2(3,4))-(DATZ2(3,3))-(DATZ2(4,4))))
3580        ZMOD5=-(CM3/CM4)
3590        DATZ3(I,J)=DATZ2(I,J)+ZMOD5
3600    216 CONTINUE
3610    215 CONTINUE
3620        DATY2(1,1)=DATAY(1,1)
3630        DATY2(1,2)=DATAY(1,2)
3640        DATY2(2,1)=DATAY(2,1)
3650        DATY2(2,2)=DATAY(2,2)
3660        DATY2(1,3)=DATAY(1,4)+DATAY(1,5)
3670        DATY2(2,3)=DATAY(2,4)+DATAY(2,5)
3680        DO 221 I=1,2
3690        DATY2(3,I)=DATAY(4,I)+DATAY(5,I)
3700    221 CONTINUE
3710        DATY2(3,3)=(DATAY(4,4)+DATAY(5,5)+(2*DATAY(4,5)))
3720 *        NOW WRITE THE OUTPUT FILE
3730    270 WRITE(LUOU,20)NPR1,NOCASE,NRC,NFGC,NCC,NTPC,NTOT,NPHSET,NSP
3740       &       ,NSN,NEUF
3750        IF(NPHASE.EQ.2)GOTO400
3760        NPM=NPHASE+1
3770        IF(NPHASE.EQ.1)DATZ3(1,1)=DATAZ(1,1)
3780        IF(NPHASE.EQ.1)DATZ3(2,1)=DATAZ(4,1)
3790        IF(NPHASE.EQ.1)DATZ3(1,2)=DATAZ(1,4)
3800        IF(NPHASE.EQ.1)DATZ3(2,2)=DATAZ(4,4)
3810        IF(NPHASE.EQ.1)DATY2(1,1)=DATAY(1,1)
3820        IF(NPHASE.EQ.1)DATY2(1,2)=DATAY(1,4)
3830        IF(NPHASE.EQ.1)DATY2(2,1)=DATAY(4,1)
3840        IF(NPHASE.EQ.1)DATY2(2,2)=DATAY(4,4)
3850        DO 210 K=1,NPM
3860        DO 220 L=1,NPM
3870        NPR1=NPR1+10
3880        WRITE(LUOU,30)NPR1,DATZ3(K,L)
3890    220 CONTINUE
3900    210 CONTINUE
3910    400 NS=1
3920        IF(NPHASE.NE.2)GOTO430
3930        DO 410I=1,NPM
3940        DO 420 J=1,NPM
```

```
3950          NPR1=NPR1+10
3960          WRITE(LUOU,30)NPR1,DATZ3(I,J)
3970       42 CONTINUE
3980       41 CONTINUE
3990       43 K=1
4000          DO 230 I=1,NPN
4010          DO 240 L=1,NPN
4020          NPR1=NPR1+10
4030          WRITE(LUOU,30)NPR1,DATY2(K,L)
4040       24 CONTINUE
4050       23 CONTINUE
4060          WRITE(LUOU,40)NFN
4070          PRINT,'TYPE A 1 TO PROCESS A NEW INPUT FILE'
4080          READ,I
4090          CALL DETACH(LUIN,ISTAT,)
4100          CALL DETACH(LUOU,ISTAT,)
4110          IF(I.EQ.1)GOTO110
4120       28 PRINT,'STOP OF EXECUTION NORMAL'
4130       99 PRINT,'END OF DATA FILE'
4140          STOP
4150          END
020           SUBROUTINE NEURED(Y,Z,N,M,NDIMZY,SA1,SA2,SA3,WKAREA,NW,NDIM)
030 *           THIS SUBROUTINE COMPUTES A NEUTRAL REDUCTION PROCEDURE
040 *           USING A GENERALIZED MATRIX APPROACH DEVELOPED BY R.RUSTAY
050 *           NOTE THAT THE RESULTS ARE RETURNED IN Z AND Y WITH ZEROED
060 *           BORDERS.  THUS ORIGINAL Z,Y, ARE DESTROYED
070           COMPLEX SA1(NDIM,NDIM),Y(NDIMZY,NDIMZY),SA3(NDIM,NDIM)
080           COMPLEX SA2(NDIM,NDIM),Z(NDIMZY,NDIMZY)
090           COMPLEX SUM,ZERO,ZONE,DEN
100           DIMENSION WKAREA(NW)
110           DATA ZERO/(0.,0.)/,ZONE/(1.,0.)/
120 *           IF M=N+1 THERE IS NO NEED FOR COMPUTATION
130           IF(M.EQ.(N+1))RETURN
140           L=N+1
150           K=M-N
160 *           REDUCTION OF Y
170           DO 210 I=1,N
180           SUM=ZERO
190           DO 200 J=L,M
200        200 SUM=SUM+Y(I,J)
210           Y(I,L)=SUM
220        210 Y(L,I)=SUM
230           SUM=ZERO
240           DO 220 I=L,M
250           DO 220 J=L,M
260        220 SUM=SUM+Y(I,J)
270           Y(L,L)=SUM
280 *           THIS NEXT STEP OF ZEROING BORDER IS NOT NECESSARY
290           CALL CMTZRO(Y,L,L,NDIMZY)
300 *           REDUCTION OF Z
310 *             EXTRACT Z22 AND Z12 AND PLACE IN SA1 AND SA2 RESPECTIVELY
320           DO 320 J=1,K
```

```
      JPN=J+N
      DO 300 I=1,K
  300 SA1(I,J)=Z(I+N,JPN)
      DO 310 I=1,N
  310 SA2(I,J)=Z(I,JPN)
  320 CONTINUE
*         NEXT INVERT Z22 USING SA1 AND COMPUTE DEN=ST*Z22I*S
      CALL CMTINV(SA1,K,K,NDIM,WKAREA)
      DEN=ZERO
      DO 330 I=1,K
      DO 330 J=1,K
  330 DEN=SUM+SA1(I,K)
*           NEXT COMPUTE Z12*Z22I (NXM=N) AND SAVE IN SA3.  WILL THEN NO
*           LONGER NEED Z22I=SA1 SO THAT SA1 WILL BECOME SCRATCH
      CALL CMTMPY(SA2,SA1,SA3,N,K,K,NDIM)
*           NOW SA3=Z12*Z22I, SA1 IS SCRATCH, SA2=Z12
*           NEXT COMPUTE SA1=Z12*Z22I*Z12T=SA3*Z12T=SA3*SA2T
      CALL CMTMPT(SA3,SA2,SA1,K,N,N,NDIM)
*           NOW SA1=Z12*Z22I*Z12T, SA3=Z12*Z22I, SA2 IS SCRATCH
*           NEXT COMPUTE Z11-Z12*Z22I*Z12T AND DO IN PLACE
      DO 350 I=1,N
      DO 350 J=1,N
  350 Z(I,J)=Z(I,J)-SA1(I,J)
*           NOW SA1 AND SA2 ARE SCRATCH,  REMEMBER SA3=Z12*Z22I
*           NEXT COMPUTE SA1=Z12*Z22I*S, IE SA1 BECOMES AN NX1 VECTOR
      DO 370 I=I,N
      SUM=ZERO
      DO 360 J=1,K
  360 SUM=SUM+SA3(I,J)
  370 SA1(I,1)=SUM
*           NOW SA1=Z12*Z22I*S AN N-VECTOR, SA2 AND SA3 ARE SCRATCH
*           NEXT COMPUTE SA2=Z12*Z22I*S*ST*Z22I*Z12T=SA1*SA1T
      CALL CMTMPT(SA1,SA1,SA2,1,N,N,NDIM)
*           NOW FORM Z11E, REMEMBERING Z11-Z12*Z22I*Z12T HAS ALREADY
*           BEEN DONE.  ALSO SA1=Z12*Z22I*S AN N-VECTOR,
*           SA2=Z12*Z22I*S*ST*Z22I*Z12T AND SA3 IS SCRATCH
      DO 420 I=1,N
      DO 400 J=1,N
  400 Z(I,J)=Z(I,J)+SA2(I,J)/DEN
*           NOW FILLING IN REMAINING PARTITIONS
      Z(I,L)=SA1(I,1)/DEN
  420 Z(L,I)=S(I,L)
      Z(L,L)=ZONE/DEN
*           ZEROING BORDER
      CALL CMTZRO(Z,L,L,NDIMZY)
      RETURN
      END
```

```
010 *       THIS PROGRAM READS ASCII DISTRIBUTION NETWORK CONFIGURATION DATA
020 *       FILES & REARRANGES THEIR FORMAT PRESENTATION.
030 *       BELOW IS AN EXPLANATION OF THE ABBREVIATIONS USED TO REPRESENT
040 *       THE NETWORK CONFIGURATION DATA IN THE READ & WRITE STATEMENTS
050 *       OF THIS PROGRAM.
060 *       ILIN=LINE NUMBER FROM NETWORK CODING FORM USED IN READ STATEMENT
070 *       NODE=NETWORK NODE NUMBER
080 *       IPAR=NETWORK PARENT NODE
090 *       ILSN=LEFT SON OF NETWORK PARENT NODE
100 *       IRSN=RIGHT SON OF NETWORK PARENT NODE
110 *       ILCD=NETWORK NODE LENGTH CODE
120 *       IOHM=OHM CODE
130 *       ITTY=TRANSFORMER TYPE CODE
140 *       IPHS=PHASE CONNECTION CODE
150 *       NSER=NUMBER OF ELECTRIC SERVICES ASSOCIATED WITH A NODE
160 *       IPRL=PRIMARY LOAD CODE
170 *       INGD=NEUTRAL = GROUND ADMITTANCE CODE
180 *       LINE=NEW NETWORK DATA LINE NUMBERS GENERATED BY THIS PROGRAM
190         CHARACTER INFILE*9,OUFILE*9
200         DATA IOHM/4/,LUIN/10/,LUOU/11/,LTYP/1000000/,INGD/0/
210         DATA INFILE/'NTAKING1;'/,OUFILE/'DNAKIN01;'/
220      10 FORMAT(V)
230      11 FORMAT(I4,4I5,1X,I4,1X,7I5)
240      12 FORMAT('0001 NODE IPAR ILSN IRSN LTYP
250                'ILCD IOHM ITTY IPHS NSER IPRL INGD')
260         CALL NASTEK
270     130 PRINT,'INFILE,OUFILE=',INFILE,' ',OUFILE
280         PRINT,'IF OK TYPE CR'
290         READ,I
300         IF(I.EQ.0)GOTO150
310         PRINT,'TYPE NAME OF INFILE,OUFILE, EACH FOLLOWED BY ;'
320         READ,INFILE,OUFILE
330         GOTO130
340     150 CONTINUE
350         CALL OPENF(LUIN,INFILE,ISTAT,3,0,1)
360         IF(ISTAT.EQ.0)GOTO160
370         IF(ISTAT.EQ.5)PRINT,INFILE,'DOES NOT EXIST'
380         STOP 0160
390     160 CALL OPENF(LUOU,OUFILE,ISTAT)
400         IF(ISTAT.EQ.0)GOTO200
410         IF(ISTAT.NE.0)PRINT,'ISTAT=',ISTAT
420         IF(ISTAT.EQ.5)PRINT,OUFILE,'ISTAT=',ISTAT
430         STOP 0300
440     200 LINE=1000
450         WRITE(LUOU,12)
460         READ(LUIN,10)ILIN,NODE
470         WRITE(LUOU,11)LINE,NODE
480         LINE=LINE+1
490     210 CONTINUE
```

```
      READ(LUI..,10,END=990)ILIN.MODE,IPAR,ILSN,IRSN,LTYP,ILCD,IOHM,
     &      ITTY,IPHS,NSER,IPRL,INGD
      WRITE(LUOU,11)LINE,MODE,IPAR,ILSN,IRSN,LTYP,ILCD,IOHM,
     &      ITTY,IPHS,NSER,IPRL,INGD
      LINE=LINE+1
      GOTO210
  990 CALL DETACH(LUIN,ISTAT,)
      CALL DETACH(LUOU,ISTAT,)
      PRINT,OUFILE,' COMPLETE'
      PRINT,'REMOVE FIRST ANNOTATION RECORD BEFORE USE'
      STOP
      END
```

```
00010** RUNH =NTWKAN4(ULIB,CORE=40K)LIBRARY4;LIBRARY/IMSL,R;
00020** LIBRARY/OLDTSLIB,R;LIBRARY/ZETA,R
00030 *       THIS IS A VERSION OF NTWKANS1 WHICH INCLUDES THE
00040 *       SUBROUTINE ZPLCTM.  BESURE TO REDEFINE NDIPLT IN DATA
00050 *       STATEMENT TO CORRESPOND TO ANY CHANGE IN DIMENSION OF XA
00060 *       THE ROOT PROGRAM HAS BEE CHANGED
00070 *       TO INCLUDE CODE TO SET UP PARAMETERS AND CALL THE SUB-
00080 *       ROUTINE.  THE SUBROUTINE ITSELF HAS BEEN ADDED TO
00090 *       LIBRARY 4.  USE J.J.FLECK1/C TO NOTE ALL DIFFERENCES.
00100 *
00110 *       THIS VERSION BY RCR INCLUDES SINGLE PHASE RATIO BANK
00120 *       TRANSFORMER LOGIC AND A REVISED PROCEDURE FOR CALCULATING
00130 *       TRANSPOSITIONS AND TRANSISTIONS
00140 *       THIS PROGRAM ANALYZES AND DETERMINES VOLTAGES AND CURRENTS
00150 *       GIVEN THE SOURCE PARAMETERS AND RESULTS OF APPROPRIATE NTWKERS
00160 *       PROGRAM CONTAINED IN APPROPRIATE FILE NT123456
00170 *
00180 *       THIS VERSION ALSO DIFFERS FROM NTWKERS1 IN THAT SEVERAL
00190 *       UNUSED VARIABLES HAVE BEEN REMOVED FROM THE CODE.  THE
00200 *       VARIABLES 'ZC', 'OIYLS', AND 'OIY2S' WERE BASICALLY UN-
00210 *       USED AND WERE THEREFORE REMOVED.
00220 *
00230 *       THE FIRST SECTION ACCESSES PROPER NT123456 FILE
00240 *
00250 *       SEE THE END OF THIS LISTING FOR PROGRAM ORGANIZTION NOTES
00260 *       AND NOMENCLATURE LIST
00270 10000 FORMAT('THIS PROGRAM ANALYZES VOLTAGES AND CURRENTS'/
00280      & 'APPROPRIATE NT    FILE MUST BE AVAILABLE IN DIRECTORY '/)
00290 10001 FORMAT('IF OK TYPE CR')
00300 10002 FORMAT(2HNT,I2,2I1,I2,1H;)
00310 10003 FORMAT('FILE ',A10,'  NOT AVAILABLE ON LGU ',I5/
00320      & 'DUE TO ERROR CONDITION ',I5/'PROGRAM ABORTS')
00330 10004 FORMAT(//'FILE RREAD ERROR ON ',I6,'FILE AT RECORD',I5/
00340      & ' PROGRAM ABORTS')
00350 10005 FORMAT(1HN,I2,2I1,I2,I1,1H;)
00360 10006 FORMAT(V)
00370 10007 FORMAT('THE FOLLOWING PARAMETERS PERTAIN TO THIS ANALYSIS'/
00380      & '  FREQUENCY CODE=',I5/' TEMPERATURE CODE=',I5/'ASSUMPTION ',
00390      & 'CODE=',I5/' NETWORK IDENTIFICATION NUMBER=',I5/
00400      & ' PROCESSING OPTION=',I5///'NETWORK DESCRIPTION :'/)
00410 10008 FORMAT('SECTION PARENT LSON RSON LEVEL LTYP LENGTH OHMCD ')
00420 10009 FORMAT(I5,I8,2I6,2I7,I8,I7)
00430 10010 FORMAT('TYPE 1 TO SPECIFY A NEW SOURCE, TYPE 2 TO EVALUATE'/
00440      & 'A NEW SECTION USING EXISTING EXCITATION  IMBEDDED IN FILE')
00450 10011 FORMAT(A50,I5/(10I10))
00460 10012 FORMAT(A50,I5/8(2X,E12.6))
00470 10013 FORMAT(//'WARNING - FREQUENCY SHIFT COMPUTATION USING ',
00480      & 'LENGTH MODIFICATION'/'MULTIFICATION FACTOR=',F9.6//)
00490 10014 FORMAT(/'ENTER VOLTAGE SOURCE VECTOR AS ',I3,' COMPLEX NUMBERS'/
```

```
00500      & ' I.E. ',I3,' REAL NUMBERS SEPERATED BY COMMAS.'/)
00510 10015 FORMAT('DO YOU WANT A VOLTAGE SOURCE DRIVE?')
00520 10016 FORMAT(//'PLEASE ENTER SOURCE CURRENT VECTOR AS ',I3,' COMPLEX',
00530      & ' NUMBERS '/'I.E. ',I3,' REAL NUMBERS,SEPERATED BY COMMAS'/)
00540 10017 FORMAT(/'ARE THE ABOVE VALUES CORRECT ? ')
00550 10018 FORMAT(/'ENTER SOURCE ADMITTANCE MATRIX AS ',I3,' COMPLEX ',
00560      & 'NUMBERS '/'I.E. ',I3,' REAL NUMBERS,SEPERATED BY COMMAS'/
00570      & 'FOR CONVENIENCE YOU MAY ENTER ',I3,' LINES OF ',I3,
00580      & ' REAL NUMBERS EACH'/' ROW ORDER ENTRIES EXPECTED'/)
00590 10019 FORMAT(/ 4(2X,E14.8))
00600 10020 FORMAT(/'ENTER A CONCATENATED INTEGER TO INDICATE OUTPUT',
00610      & ' OPTIONS AS FOLLOWS'/' 1 - OUTPUT VOLTAGE & INPUT CURRENT'/
00620      & '2 - NORTON SOURCE ADMITTANCE AND CURRENT'/'3 - INPUT',
00630      & ' ADMITTANCE'/'4 - TOTAL LOAD ON SECTION'/'5 - VOLTAGE TRANSFER',
00640      & ' MATRIX'/'6 - CURRENT DRIVE FOR UNITY VOLTAGE'/
00650      & '7 - OVERALL VOLTAGE TRANSFER'/
00660      & '9 - MAGNITUDES OF OUTPUT VOLTAGES AND IMPEDANCE',
00670      & ' LEVELS, LEVEL WARNINGS AND POWER'/'THE LAST OPTION WILL',
00680      & ' AFTER THE FIRST PRINTOUT,'/'DISAPLE ALL OTHER OPTIONS'//)
00690 10021 FORMAT(/A50/(8(2X,E12.6)))
00700 10022 FORMAT(A50/(4(2X,E14.8)))
00710 10023 FORMAT(/A50/8(2X,E 12.6))
00720 10024 FORMAT(//'FROM NOW ON ONLY VOLTAGES AND POWER ARE ',
00730      & 'WRITTEN'/'ALL OTHER PRINTOUTS HAVE BEEN SUPPRESSED'/
00740      & ' SEC  SIR  PAR CUR DIST  /V1/    /V2/    /V3/  ZTR/MIN VFZF',
00750      & '  YINMX   YSIBMX  YNORMX SIB/* SIB/N   POWER'/)
00760 10025 FORMAT(3I5,F8.1,3(1X,F7.4),1X,F7.2,2A2,3(1X,F7.2),2I5,1PE14.4)
00770 10026 FORMAT(/' NREAD ',I5,'    SECTION NUMBER ',I3,'   LINE TYPE ',I5/
00780      & '  LCU ',I2,'     IRECO ',I2,'         IAREC ',I5,'  NECO ',I3/)
00790 10027 FORMAT('ENTER SECTION NUMBER OF SECTION AT WHICH OUTPUT ',
00800      & 'IS DESIRED')
00810 10028 FORMAT('SECTION CHOSEN IS ',I6,' IS THIS CORRECT?')
00820 10029 FORMAT(' SECTION  ',I3,'  WHICH YOU HAVE CHOSEN IS NOT IN  '
00830      & 'NETWORK ',I4/)
00840 10030 FORMAT(' YOU HAVE EITHER FAILED TO DEFINE A SOURCE OR'/
00850      & 'A PROCESSING ERROR ON SECTION ',I5,'   HAS TAKEN PLACE'/
00860      & '    PROGRAM ABORTS')
00870 10031 FORMAT(/A50/(8I8))
00880 10032 FORMAT(/'DO YOU WANT OUTPUT AT ANOTHER SECTION WITH SAME SOURCE?')
00890 10033 FORMAT(/'DO YOU WANT THE EFFECTS OF LOAD PERTURBATION '/
00900      & 'OF A REVERSE PATH PROPOGATION ?')
00910 10034 FORMAT(/'DO YOU WISH TO REANALYZE SAME NETWORK WITH A '/
00920      & ' DIFFERENT SOURCE ?')
00930 10035 FORMAT(/'DO YOU WISH TO ANALYZE ANOTHER NETWORK ? ')
00940 10036 FORMAT(/'ENTER SECTION AT WHICH LOAD CHANGE TAKES PLACE'/)
00950 10037 FORMAT('ENTER NORTON SOURCE CURRENT VECTOR AS'/
00960      & I5,' COMPLEX NUMBERS'/)
00970 10038 FORMAT('ENTER CHANGEIN NORTON ADMITTANCE AS'/
00980      & I8,' COMPLEX NUMBERS'/)
00990 10039 FORMAT(/'IS THIS A REVERSE PATH PROPOGATION ?')
01000 10040 FORMAT(/'ENTER CHANGE IN LOAD DLY2=YLNEW-YLOLD AS'
01010      & ,I5,'COMPLEX NUMBERS'/'FOR CONVENIENCE YOU MAY'
```

```
01020       & 'ENTER',I3,'LINES OF ',I3,'COMPLEX NUMBERS EACH')
01030 10041 FORMAT(/'ENTER SECTION AT WHICH EFFECT OF PERTURBATION IS '
01040       & 'TO BE ENTER ',I3)
01050 10042 FORMAT('TWO SEPERATE NETWORKS INDICATED PROGRAM ABORTS')
01060 10043 FORMAT(/'VOLTAGE BEFORE CHANGE')
01070 10044 FORMAT(/'CHANGE IN VOLTAGE')
01080 10045 FORMAT(/'VOLTAGE AFTER CHANGE')
01090 10046 FORMAT(/'DO YOU WANT RESULT AT ANOTHER SECTION?')
01100 10047 FORMAT(/'DO YOU WANT ANOTHER PERTURBED LOAD?')
01110       IMPLICIT COMPLEX(A-G,O,S,T,V,Y,Z)
01120       CHARACTER *9 MDPRL,MDOUT
01130       CHARACTER *8  JDATE
01140       CHARACTER *2 JVOLT,JIMPED
01150       CHARACTER *30 IXC,IYC ITI
01160       DATA LDPRL/10/LDOUT/12/IHEAD/0/NPERT/0/IBRAN/0/
01170       DATA NDPRL/1800/NSIZE/1800/NDIPLT/200/NDIM/4/IALL/0/
01180       DIMENSION VSORCE(4),Y1M(4,4),Y2K(4,4),YLOAD(4,4),YNORPH(4,4),
01190      & YTEMP(4,4),
01200      & INLCSA(1800,14),INSEQV(1800),AM(4),BM(4),CM(4),DM(4),VM(4),
01210      & CJDRIV(4),
01220      & APHI(4,4),BPHI(4,4),CPHI(4,4),DPHI(4,4),VTOPH(4,4),VPHIN(4),
01230      & VPHOUT(4),POV(20),
01240      & CUPHI(4),CNPHI(4),PRT(10),
01250      & WKAREA(200),CSDATA(4),
01260      & Z12(4,4),ZTOT(4,4),CUPTP(4),DLY2(4,4),DELV2(4),DELV1(4),
01270      & TTRAN(4,4),YSIB(4,4),
01280      & VTTOTL(4,4),VTTMP(4,4),ZIN(4,4),YTEMP2(4,4),CUREV(4),VSUM(4),
01290      & DELV3(4),YTEMP1(4,4),
01300      & ZTEMP(4,4),YOUT(4,4),YSORPH(4,4),VOUTMP(4),CSPHI(4),XA(200,4),
01310      & YSDATA(4,4),
01320      & LPRINT(9),ISTAK(1800)
01330       DATA ZERO/(0.0,0.0)/AONE/(1.0,0.0)/
01340       CALL NASTRK
01350       ITEMP=6
01360       IASN=4
01370       PVLIM=0.1
01380       PILIM=50.0
01390       DO 10 I=1,4
01400       VSORCE(I)=AONE
01410       CSPHI(I)=AONE
01420       DO 10 J=1,4
01430       YSDATA(I,J)=ZERO
01440    10 CONTINUE
01450    20 WRITE(42,10000)
01460       JA=1
01470       NPER=1
01480       IPRFLG=0
01490       IPLTF=0
01500       IPFLG=0
01510       NECMX=4
01520    30 PRINT,'TEMPERATURE =',ITEMP,' ASSUMPTION CODE=',IASN
01530       WRITE(06,10001)
```

```
01540        READ,ITEST
01550        IF(ITEST.LE.0)GOTO40
01560        PRINT,'ENTER TYPE TEMP CODE,ASSUMPTION CODE - EACH 1 DIGIT'
01570        READ,ITEMP,IASN
01580        GOTO30
01590     40 PRINT,'ENTER FREQ CODE,NETWORK ID NO.,OUTPUT FILE NO.'
01600        READ,IFCODE,NIDEN,IOPTN
01610        PRINT,'IFCODE,NIDEN,IOPTN=',IFCODE,NIDEN,IOPTN
01620        WRITE(06,10001)
01630        READ,ITEST
01640        IF(ITEST.NE.0)GOTO40
01650        PRINT,'TYPE CR TO BYPASS PLOTTING LOGIC'
01660        READ,IPFLG
01670     50 PRINT,'VOLT LIMIT =',PVLIM,' IMPEDANCE LIMIT',PILIM
01680        WRITE(06,10001)
01690        READ,ITEST
01700        IF(ITEST.EQ.0)GOTO60
01710        PRINT,'ENTER VOLTAGE AND IMPEDANCE LIMITS'
01720        READ,PVLIM,PILIM
01730        GOTO50
01740  *       NEXT WE OPEN APPROPRIATE MT     FILE AND VERIFY ITS HEADER
01750  *       WE ALSO OPEN A SEQUENTIAL OUTPUT FILE
01760     60 CALL NASTRK
01770        ENCODE(MDPRL,10002)IFCODE+1000,ITEMP,IASN,NIDEN
01780        CALL OPENF(LDPRL,MDPRL,ISTAT,3,1,1)
01790        IF(ISTAT.EQ.0)GOTO70
01800        WRITE(42,10003)MDPRL,LDPRL,ISTAT
01810        STOP 20
01820     70 CALL RANSIZ(LDPRL,MDPRL,1)
01830        READ(LDPRL'1,ERR=90)JIDEN,JDATE,XTIME,JFC,JTM,
01840       & JASN,NOMAX,MREC,NLCSA,NSEQV,NBOAT,MAXND,NECO,JROOT,PMULT,IOPC,
01850       & XDB,XANG
01860        IF(NIDEN.NE.JIDEN)CALL FILSTO(MDPRL,'NETWORK ID',NIDEN,JIDEN,610)
01870        IF(IFCODE.NE.JFC)CALL FILSTO(MDPRL,'FREQ CODE',IFCODE,JFC,620)
01880        IF(ITEMP.NE.JTM)CALL FILSTO(MDPRL,'TEMP CODE',ITEMP,JTM,630)
01890        IF(IASN.NE.JASN)CALL FILSTO(MDPRL,'ASSUM CODE',IASN,JASN,635)
01900        IF(MAXND.NE.NSIZE)CALL FILSTO(MDPRL,'DIMENSION',NSIZE,MAXND,650)
01910  *       SAVING NECO FROM HEADER FOR NEW SOURCE MODE
01920        MECO=NECO
01930        JRLCU=NECO
01940        GOTO90
01950     80 WRITE(42,10004)LDPRL,JA
01960        STOP 30
01970  *       NEXT WE READ INLCSA+INSEQV
01980     90 DO 100 JA=2,15
01990        READ(LDPRL'JA,ERR=80)(INLCSA(NINDX,JA-1),NINDX=1,MAXND)
02000    100 CONTINUE
02010        JA=18
02020        READ(LDPRL'JA,ERR=80)(INSEQV(NINDX),NINDX=1,MAXND)
02030  *       NEXT CREATE SEQUENTIAL OUTPUT FILE
02040        ENCODE(MDOUT,10005)IFCODE+1000,ITEMP,IASN,NIDEN,IOPTN
02050        CALL OPENF(LDOUT,MDOUT,ISTAT)
```

```
02060       IF(ISTAT.GE.0)WRITE(42,10003) DOUT,LDOUT,ISTAT
02070 *       OUTPUT FILE HAS BEEN MADE
02080       WRITE(LDOUT,10007)IFCONF,ITEMP,IASP,NIDEN,IOPTN
02090       IF(ABS(PMULT-1.0).LT.1E-7)GOTO110
02100       WRITE(06,10013)PMULT
02110       WRITE(LDOUT,10013)PMULT
02120   110 CONTINUE
02130 *     WRITE(LDOUT,10008)
02140 *     DO 115 I=1,NMAXD
02150 *     IF(INLCSA(I,1).LT.0)GOTO115
02160 *     WRITE(LDOUT,10009)(INLCSA(I,J),J=1,8)
02170 * 115 CONTINUE
02180 *       LET USER CHOOSE BETWEEN SPECIFYING  NEW SOURCE OR
02190 *       USING EXISTING SOURCE (IMBEDDED IN NT     FILE) TO EVALUATE
02200 *       A NEW DESIRED NSECT
02210   120 WRITE(42,10010)
02220       READ,ITEST
02230       IF(ITEST.EQ.1)GOTO190
02240       IF(ITEST.EQ.2)GOTO130
02250       GOTO120
02260 *       RECREATE OLD SOURCE FROM NT123456 FILE
02270   130 READ(LDPRL'16)JROOT,YMORPH,YTEMP,Z12,VOUTMP,CSPHI
02280       DO 140 I=1,NECO
02290       CSDATA(I)=CSPHI(I)
02300       DO 140 J=1,NECO
02310       YSDATA(I,J)=YMORPH(I,J)
02320   140 CONTINUE
02330       ITEST=5
02340       GOTO240
02350 *       THIS SECTION PERMITS USER TO SPECIFY SOURCE
02360 *       AT PRESENT ONLY CURRENT SOURCE AND SOURCE ADMITTANCE ARE
02370 *       PERMITTED,OTHER TYPES MAY BE INCLUDED LATER
02380 *       FOR PRESENT SOURCE PARAMETERS MUST BE INTRODUCED FROM
02390 *       TERMINAL.  INPUT FROM FILES MAY BE DEVELOPED LATER
02400 *
02410 *       THIS PART IS USED FOR  IDEAL VOLTAGE ENTRY ONLY
02420 *       IT HAS BEEN ADDED LATER AND CAN BE RECOGNIZED BY THE 95X LINE NUMB
02430   150 WRITE(42,10014)NECO,2*NECO
02440       READ,(VSORCE(I),I=1,NECO)
02450   160 CALL CVEPRT(VSORCE,NECO,NDIM,42)
02460       IVSORC=1
02470       WRITE(42,10001)
02480       READ,ITEST
02490       IF(ITEST.NE.0)GOTO150
02500   170 PRINT,'THE  SOURCE VECTOR WILL CREATED WITHIN'
02510       PRINT,'THE PROGRAM BY CURRENT INJECTION.   IF YOU DESIRE A'
02520       PRINT,'ZERO EQUIVALENT NORTONS SOURCE ADMITTANCE TYPE CR'
02530       READ,ITEST
02540       IF(ITEST.NE.0)GOTO220
02550 *       SETTING YSDATA=0 FOR ZERO ADMITTANCE VOLTAGE DRIVE
02560       DO 180 I=1,NECO
02570       DO 180 J=1,NECO
```

```
02580         YSDATA(I,J)=ZERO
02590     100 CONTINUE
02600         GOTO240
02610 *         RESTORING NECO TO SOURCE VALUE FOR POSSIBLITY
02620 *         THAT A NEW SOURCE IS TO BE SPECIFIED
02630     140 NECO=MECO
02640         WRITE(42,10015)
02650         WRITE(42,10001)
02660 *       NOTE IVSORC IS NOT RESET ANYWHERE
02670         READ,IVSORC
02680         IF(IVSORC.EQ.0)GOTO160
02690         IVSORC=0
02700         GOTO210
02710     200 WRITE(42,10016)NECO,2*NECO
02720         READ,(CSPHI(I),I=1,NECO)
02730     210 CALL CVEPRT(CSPHI,NECO,NDIM,42)
02740         WRITE(42,10001)
02750         READ,ITEST
02760         IF(ITEST.NE.0)GOTO200
02770         GOTO170
02780     220 WRITE(42,10018)NECO*NECO,2*NECO*NECO,NECO,2*NECO
02790         READ,((YSDATA(I,J),J=1,NECO),I=1,NECO)
02800         CALL CMTPRT(YSDATA,NECO,NDIM,42)
02810     230 CONTINUE
02820         WRITE(42,10001)
02830         READ,ITEST
02840         IF(ITEST.NE.0)GOTO220
02850     240 CONTINUE
02860         DO 250 I=1,NECO
02870         DO 250 J=1,NECO
02880         YSORPH(I,J)=YSDATA(I,J)
02890     250 CONTINUE
02900         IF(ITEST.EQ.5)GOTO270
02910 *         NEXT WE INITIALIZE ALL PERTINENT PSW TO HAVE
02920 *         A LEAST SIGNIFICANT DIGIT OF 1
02930 *         REMEMBER INLCSA(I,13)ALSO CONTAINS PRIMARY LOAD TYPE INFO
02940         DO 260 I=1,MAXND
02950         IF(INLCSA(I,1).LT.0)GOTO260
02960         JTENS=INLCSA(I,13)/10
02970         INLCSA(I,13)=10*JTENS+1
02980     260 CONTINUE
02990     270 CONTINUE
03000 *       DO 301 IN=1,14
03010 *       WRITE(LDOUT,10011)'INLCSA COLUMN',IN,(INLCSA(IP,IN),IP=1,160)
03020 *   301 CONTINUE
03030 *       WRITE(LDOUT,10011)'INSEQV',1,(INSEQV(IP),IP=1,80)
03040 *         NEXT DETERMINE WHAT SECTION INFORMATION IS DESIRED TO BE
03050 *         PRINTED OUT
03060 *         FOR OPTIONS REFER NOTES
03070 *         IPRINT IS A CONCATENATED NUMBER
03080         WRITE(42,10020)
03090         READ,IPRINT
```

```
03100         I DFLT=0
03110 *         PARSING OF IPRINT
03120         DO 290 I=1,3
03130         LPRINT(I)=0
03140     290 CONTINUE
03150     290 JPRT=IPRINT/10
03160         KPR=IPRINT-10*JPRT
03170         IF(KPR.NE.0)LPRINT(KPR)=1
03180         IF(JPRT.EQ.0)GOTO300
03190         IPRINT=JPRT
03200         GOTO290
03210 *         NEXT WE DETERMINE INPUT VOLTAGE
03220     300 IF(ITEST.EQ.5)GOTO610
03230         IAREC=INLCSA(JROOT,14)
03240 *       WRITE(42,10011)'LPRINT VECTOR',JROOT,LPRINT
03250         LTYP=INLCSA(JROOT,6)
03260         NO=JROOT
03270         NSECT=JROOT
03280 *       WRITE(42,10031)'ROOT SECTION',NSECT,IAREC,LTYP,0,NECO
03290         ASSIGN 310 TO NEXSTA
03300         GOTO550
03310     310 CONTINUE
03320 *         NECO FOR ROOT SECTION HAS BEEN DEFINED IN HEADER RECORD
03330 *         OF NT      FILE
03340         IF(LCU.NE.NECO)STOP 0080
03350         DO 320 I=1,LCU
03360         DO 320 J=1,LCU
03370         YTEMP(I,J)=YIN(I,J)+YSORPH(I,J)
03380         Z12(I,J)=YTEMP(I,J)
03390     320 CONTINUE
03400 *         AT THIS POINT YIN BECOMES AVAILABLE AND WE CAN COMPUTE
03410 *         CURRENT DRIVE TO YIELD DESIRED VOLTAGE AT THE INPUT
03420         IF(IVSORC.EQ.1)CALL CMVMPY(YTEMP,VSORCE,CSPHI,LCU,LCU,
03430        &                   NDIM,NDIM,NDIM)
03440 *         NEXT WE ENTER SOURCE PARAMETERS INTO OUTPUT FILE
03450         WRITE(LDOUT,10021)'SOURCE CURRENT'
03460         CALL CVEPRT(CSPHI,NECO,NDIM,LDOUT)
03470         WRITE(LDOUT,10021)'SOURCE ADMITTANCE'
03480         CALL CMTPRT(YSDATA,NECO,NDIM,LDOUT)
03490         NPAR=0
03500         NSIB=0
03510         DO 330 I=1,NDIM
03520         DO 330 J=1,NDIM
03530     330 YSIB(I,J)=ZERO
03540         WRITE(LDOUT,10012)'Y INPUT',NSECT
03550         CALL CMTPRT(YIN,NECO,NDIM,LDOUT)
03560         WRITE(LDOUT,10012)'Y SOURCE+YIN',NSECT
03570         CALL CMTPRT(YTEMP,NECO,NDIM,LDOUT)
03580         CALL CMTINV(Z12,LCU,LCU,NDIM,WKAREA)
03590 *         THE FOLLOWING TWO PAIR LOOPS COULD BE COMBINED FOR
03600 *         MORE EFFICIENCY
03610         DO 340 I=1,LCU
```

```
03620         VOUTMP(I)=ZERO
03630         CSPHI(I)=ZERO
03640         CSPHI(I)=CSPHI(I)
03650         DO 340 J=1,LCU
03660         VOUTMP(I)=VOUTMP(I)+Z12(I,J)*CSPHI(J)
03670         VPHIN(I)=VOUTMP(I)
03680 *       YSORPH(I,J)=YSORPH(I,J)
03690    340  CONTINUE
03700         DO 350 I=1,NDIM
03710         VPHOUT(I)=ZERO
03720         DO 350 J=1,LCU
03730         CSPHI(I)=CSPHI(I)+YIN(I,J)*VOUTMP(J)
03740         IF(I.GT.NECO)GOTO350
03750         VPHOUT(I)=VPHOUT(I)+VTOPH(I,J)*VOUTMP(J)
03760    350  CONTINUE
03770         WRITE(LDOUT,10022)'INPUT VOLTAGE'
03780         CALL CVPRT(VOUTMP,LCU,NDIM,LDOUT)
03790         WRITE(LDPRL'16)JROOT,YSORPH,YTEMP,Z12,VOUTMP,CSPHI
03800 *         INITIALIZE OVERALL VOLTAGE TRANSFER MATRICES VTTOTL,VTTMP TO IDENT
03810         DO 370 I=1,NDIM
03820       -   DO 360 J=1,NDIM
03830         VTTOTL(I,J)=ZERO
03840    360  VTTMP(I,J)=ZERO
03850         VTTOTL(I,I)=ONE
03860    370  VTTMP(I,I)=ONE
03870 *         DETERMINE PARAMETERS OF ROOT SECTION
03880 *         IN THE FOLLOWING YSORPH AND YTEMP ARE TEMPORY MATRICES
03890 *         WHICH STORE Y2K AND YOUT OF PREVIOUS SECTION ALSO.
03900 *           Z12 IS DEFINED (USING RCR FIGURE E-4D) BY
03910 *           E1=Z12*(-I2)FOR I1=0.  USING FIGURE E-4D
03920 *           Z12=(Y1+YIN)INVERSE*(D+Y1*B)-B
03930 *           YOUT=(D+YSORPH*B)INVERSE*(C+YSORPH*A)
03940 *         ARRIVE AT FOLLOWING STATEMENT ONLY FOR ROOT SECTION
03950    380  CALL CMTADD(YIN,YSORPH,YTEMP,LCU,NDIM)
03960         CALL CMABSC(DPHI,YSORPH,BPHI,YSORPH,LCU,LCU,LCU,NDIM)
03970         CALL CMTINV(YTEMP,LCU,LCU,NDIM,WKAREA)
03980         CALL CMTMPY(YTEMP,YSORPH,Z12,LCU,LCU,LCU,NDIM)
03990         CALL CMTSUB(Z12,BPHI,Z12,LCU,NDIM)
04000         CALL CMTINV(YSORPH,LCU,LCU,NDIM,WKAREA)
04010         CALL CMABSC(CPHI,YSORPH,APHI,YTEMP,LCU,LCU,LCU,NDIM)
04020         CALL CMTMPY(YSORPH,YTEMP,YOUT,LCU,LCU,LCU,NDIM)
04030         CALL CMTMPY(VTOPH,VTTMP,VTTOTL,NECO,LCU,JRLCU,NDIM)
04040         XLENTH=FLOAT(INLCSA(JROOT,7))/10.0/PMULT
04050 *         NEXT WRITE SELECTED QUANTITIES INTO OUTPUT FILE & LDPRL
04060 *         QUANTITIES SELECTED BY LPRINT
04070         ASSIGN 610 TO NEXSTA
04080 ********************************************************************
04090 *                                                                  *
04100 *      BEGIN  OUTPUT  MACRO-SECTION                                *
04110 *                                                                  *
04120 ********************************************************************
04130    390  DO 400 I=1,NDIM
```

```
04140         C[illegible]
04150         [illegible] J=1, [illegible]
04160         [illegible]
04170         IF([illegible]) GOTO 400
04180         ZTOT(I,J)=Y[illegible]+Y2K(I,J)
04190         C[illegible]IV(I)+ZTOT(I,J)
04200    400  CONTINUE
04210         CALL C[illegible](ZTOT,[illegible],LCO,NDIM,[illegible]AREA)
04220         JVOLT=' '
04230         JIMPED=JVOLT
04240         DO 410 I=1,NDIM
04250         J=NDIM
04260         POV(I)=CABS(VPHOUT(I))
04270         POV(J+I)=CABS(ZTOT(I,I))
04280         POV(2*J+I)=CABS(CURPHI(I))
04290         IF(POV(I).LT.PVLIM.AND.I.LE.NPECO)JVOLT='*'
04300         IF(POV(J+I).LT.PILIM.AND.I.LE.NPLCO)JIMPED='*'
04310    410  CONTINUE
04320         POWER=0.0
04330         DO 420 I=1,LCO
04340         POWER=POWER+REAL(VPHIN(I)*CONJG(CURPHI(I)))
04350    420  CONTINUE
04360         DO 430 I=1,NDIM
04370         VOUTMP(I)=ZERO
04380         DO 430 J=1,NDIM
04390         YTEMP(I,J)=ZERO
04400         YSORPH(I,J)=ZERO
04410    430  CONTINUE
04420         CALL CMTCPY(VTTOTL,VTTMP,JRLCO,NECO,NDIM)
04430         DO 440 I=1,NECO
04440         VOUTMP(I)=VPHOUT(I)
04450         DO 440 J=1,NECO
04460         YTEMP(I,J)=YOUT(I,J)
04470         YSORPH(I,J)=Y2K(I,J)
04480    440  CONTINUE
04490         DO 450 I=1,10
04500    450  PRT(I)=0
04510         IPRT1=0
04520         IPRT2=0
04530         IPHSE=IALCSA(ISECT,9)/1000
04540         PRT(1)=CABS(ZTOT(1,1))
04550         DO 460 I=1,LCO
04560         IF(CABS(ZTOT(I,I)).LT.PRT(1))PRT(1)=CABS(ZTOT(I,I))
04570         DO 460 J=1,LCO
04580         IF(CABS(YIN(I,J)).GT.PRT(2))PRT(2)=CABS(YIN(I,J))
04590         IF(CABS(YSIB(I,J)).GT.PRT(3))PRT(3)=CABS(YSIB(I,J))
04600         IF(CABS(YNORPH(I,J)).GT.PRT(4))PRT(4)=CABS(YNORPH(I,J))
04610    460  CONTINUE
04620         IF(PRT(2).GT.1.E-10)PRT(2)=1./PRT(2)
04630         IF(PRT(3).GT.1.E-10)PRT(3)=1./PRT(3)
04640         IF(PRT(4).GT.1.E-10)PRT(4)=1./PRT(4)
04650         PRAT1=0.0
```

```
04660          PRAT2=0.0
04670          IF(PRT(3).LT.1.E-10)GOTO470
04680          PRAT1=PRT(3)/PRT(2)
04690          PRAT2=PRT(3)/PRT(1)
04700          IF(PRAT1.LT.1.0)PRAT1=1./PRAT1
04710          IF(PRAT2.LT.1.0)PRAT2=1./PRAT2
04720    470   IPRT1=PRAT1
04730          IPRT2=PRAT2
04740  *         NOW WE WRITE OUTPUT AND LDPRL FILES AND
04750  *         CHANGE STATUS DIGIT BY ADDING 1 IF IT IS NOT ALREADY 2
04760          LSDIG=IABS(LCSA(ND,13)-IABS(LCSA(ND,13)/10*10
04770          IF(IPHSE.NE.0.AND.IPHSE.LE.3)PRT(1)=CABS(ZTOT(IPHSE,IPHSE))
04780          IF(LSDIG.LT.2)LCSA(ND,13)=LCSA(ND,13)+1
04790          GOTO(480,490,500,510),IREAD
04800  *         WRITE STATEMENTS
04810    480   WRITE(LDPRL'IAREC)NSECT,LTYP,LCU,NRECO,XLENTH,VTTOTL,DPHI,
04820         &                    VTOPH,YIN,YNORPH,
04830         &                    Y2K,YOUT,VPHIN,VPHOUT,CUPHI,CNPHI
04840          GOTO520
04850    490   WRITE(LDPRL'IAREC)NSECT,LTYP,LCU,NRECO,VTOPH,YIN,Y2K,
04860         &                    YOUT,VTTOTL,YFBTR,ZPBTR,XN,
04870         &                    YNORPH,VPHIN,VPHOUT,CUPHI,CNPHI
04880          GOTO520
04890    500   WRITE(LDPRL'IAREC)NSECT,LTYP,LCU,NRECO,XLENTH,VTTOTL,AM,BM,CM,
04900         &    APHI,BPHI,CPHI,DPHI,VTOPH,YLOAD,YOUT,
04910         &    YIN,YNORPH,Y2K,VPHIN,VPHOUT,CUPHI,CNPHI,OLOAD,VSPRAT
04920          GOTO520
04930    510   WRITE(LDPRL'IAREC)NSECT,LTYP,LCU,NRECO,XLENTH,VTTOTL,APHI,
04940         &     BPHI,CPHI,DPHI,VTOPH,YIN,YOUT,
04950         &     YNORPH,Y2K,VPHIN,VPHOUT,CUPHI,CNPHI,XL12,XL23
04960  *         AFTER THIS WE ALWAYS WRITE OUTPUT FILE
04970    520   CONTINUE
04980          IF(IPFLG.EQ.0)GOTO524
04990          INDPLT=INDPLT+1
05000          IF(INDPLT.LE.NDIPLT)GOTO522
05010          IF(IPLTF.GT.0)GOTO524
05020          IPLTF=1
05030          PRINT,'LIMITING NUMBER OF PLOT ITEMS TO NDIPLT=',NDIPLT
05040          GOTO524
05050    522   CONTINUE
05060          XA(INDPLT,1)=XLENTH
05070          XA(INDPLT,2)=POV(1)
05080          XA(INDPLT,3)=POV(2)
05090          XA(INDPLT,4)=POV(3)
05100    524   CONTINUE
05110          IF(IPRFLG.EQ.1)GOTO530
05120          IF(LPRINT(1).EQ.1)WRITE(LDOUT,10012)'OUTPUT VOLTAGE',NSECT,
05130         &                    (VPHOUT(I),I=1,NRECO)
05140          IF(LPRINT(1).EQ.1)WRITE(LDOUT,10012)'INPUT CURRENT',NSECT,
05150         &                    (CUPHI(I),I=1,LCU)
05160          IF(LPRINT(2).EQ.1)WRITE(LDOUT,10012)'NORTON ADMITTANCE',NSECT
05170          IF(LPRINT(2).EQ.1)CALL CMTPRT(YNORPH,LCU,NDIM,LDOUT)
```

```
05180       IF(LPRINT(2).EQ.1)WRITE(LDOUT,10012)'NORTON CURRENT',NSECT,
05190      &                    (CNPHI(I),I=1,LCU)
05200       IF(LPRINT(3).EQ.1)WRITE(LDOUT,10012)'INPUT ADMITTANCE',NSECT
05210       IF(LPRINT(3).EQ.1) CALL CMTPRT(YI,LCU,NDIM,LDOUT)
05220       IF(LPRINT(4).EQ.1)WRITE(LDOUT,10012)'TOTAL LOAD',NSECT
05230       IF(LPRINT(4).EQ.1)CALL CMTPRT(Y2K,NECO,NDIM,LDOUT)
05240       IF(LPRINT(5).EQ.1)WRITE(LDOUT,10012)'VOLTAGE TRANSFER',NSECT
05250       IF(LPRINT(5).EQ.1)CALL CMTPRT(VTOPH,LCU,NDIM,LDOUT)
05260       IF(LPRINT(6).EQ.1)WRITE(LDOUT,10012)
05270      &                    'CURRENT DRIVE VECTOR FOR UNITY VOLTAGE',NSECT
05280       IF(LPRINT(6).EQ.1)CALL CVEPRT(CUDRIV,NECO,NDIM,LDOUT)
05290       IF(LPRINT(7).EQ.1)WRITE(LDOUT,10012)'OVERALL VOLTAGE ',
05300      &                    'TRANSFER',NSECT
05310       IF(LPRINT(7).EQ.1)CALL CMTPRT(VTTOTL,JRLCU,NDIM,LDOUT)
05320   530 IF(LPRINT(9).EQ.1)IPRFLG=1
05330       IF(LPRINT(9).NE.1)GOTO540
05340       IF(LDOUT.EQ.42)IHEAD=0
05350       IF(IALL.NE.0.AND.LDOUT.EQ.42)IHEAD=1
05360       IF(IHEAD.NE.1)WRITE(LDOUT,10024)
05370       IHEAD=1
05380       IF(IALL.EQ.0.OR.LDOUT.NE.42)WRITE(LDOUT,10025)NSECT,NSIB,
05390      &    NPAR,XLENTH,(POV(I),I=1,3),
05400      &    PRT(1),JVOLT,JIMPED,(PRT(I),I=2,4),IPRT1,IPRT2,POWER
05410   540 CONTINUE
05420 *       RESERVED FOR OTHER OPTIONS
05430       GOTONEXSTA,(610,730,850,860)
05440 ************************************************************************
05450 *                                                                      *
05460 *              END  OUTPUT  MACRO-SECTION                              *
05470 *                                                                      *
05480 ************************************************************************
05490 *
05500 ************************************************************************
05510 *                                                                      *
05520 *              BEGIN  READ  MACRO-SECTION                              *
05530 *                                                                      *
05540 ************************************************************************
05550 *          STRATEGY ON LCU AND NRECO.
05560 *            LCU    IS THE NUMBER OF EFFECTIVE CONDUCTORS ASSOCIATED
05570 *                   WITH INPUT TO NEW SECTION
05580 *            NRECO IS THE NUMBER OF EFFECTIVE CONDUCTORS ASSOCIATED
05590 *                   WITH OUTPUT FROM NEW SECTION
05600 *            HENCE (CURRENT)NECO SHOULD EQUAL NEW LCU
05610 *
05620   550 IF(LTYP.GE.001.AND.LTYP.LE.032)NREAD=1
05630       IF(LTYP.GE.033.AND.LTYP.LE.099)NREAD=2
05640       IF(LTYP.GE.100.AND.LTYP.LE.999)NREAD=3
05650       IF(LTYP.GE.1000)                NREAD=4
05660 *
05670 *                                 NREAD
05680 *            1<=LTYP<=9             1          3:1 TRANSITION
05690 *           10<=LTYP<=23            1          THREE PHASE TRANSPOSITION
```

```
05700 *           24<=LTYP<=32          1          3:2 AND 2:1 TRANSITION
05710 *           33<=LTYP<=99          2          RATIO BANK TRANSORMER
05720 *          100<=LTYP<=999         3          NORMAL LINES
05730 *         1000<=LTYP              4          SUBNETWORKS (NOT IMPLEMENTED)
05740 *
05750       GOTO(560,570,580,590), IREAD
05760 *       1<=LTYP<=32
05770   560 READ(LDPRL'IAREC,ERR=80)KODE,LTYP,LCU,NRECO,XLENTH,VTTOTL,
05780      &                            DPHI,VTOPH,YIN,YNORPH,
05790      &                            Y2K,YOUT,VPHIN,VPHOUT,CUPHI,CNPHI
05800       GOTO600
05810 *       33<=LTYP<=99
05820   570 READ(LDPRL'IAREC,ERR=80)KODE,LTYP,LCU,NRECO,VTOPH,YIN,Y2K,
05830      &                      YOUT,VTTOTL,YRBTR,ZRBTR,XN,
05840      &                      YNORPH,VPHIN,VPHOUT,CUPHI,CNPHI
05850       GOTO600
05860 *       100<=LTYP<=999
05870   580 READ(LDPRL'IAREC,ERR=80)KODE,LTYP,LCU,NRECO,XLENTH,VTTOTL,
05880      &    AN,BN,CN,APHI,BPHI,CPHI,DPHI,VTOPH,YLOAD,YOUT,
05890      &    YIN,YNORPH,Y2K,VPHIN,VPHOUT,CUPHI,CNPHI,QLOAD,VSPRAT
05900       GOTO600
05910 *       1000<=LTYP
05920   590 READ(LDPRL'IAREC,ERR=80)KODE,LTYP,LCU,NRECO,XLENTH,VTTOTL,
05930      &        APHI,BPHI,CPHI,DPHI,VTOPH,YIN,
05940      &        YOUT,YNORPH,Y2K,VPHIN,VPHOUT,CUPHI,CNPHI,XL12,XL23
05950   600 CONTINUE
05960 *     WRITE(LDOUT,10026)NREAD,KODE,LTYP,LCU,NRECO,IAREC,NECO
05970       GOTONEXSTA,(310,710,750,770,950,1070,1220,1320,1340,1380)
05980 *
05990 ************************************************************************
06000 *                                                                      *
06010 *              END OF READ MACRO-SECTION                               *
06020 *                                                                      *
06030 ************************************************************************
06040 *
06050 *         THE FOLLOWING IS ASSOCIATED WITH SOURCE AND ROOT SECTION
06060 *         CALCULATION AND SHOULD NEVER INVOLVE 3 TO 1 PHASE
06070 *         TRANSITION OR RATIO BANK EITHER OF WHICH WILL
06080 *         HAVE UNEQUAL LCU AND NECO
06090 *         THE FOLLOWING STATEMENT IS REACHED AFTER SETTING UP
06100 *         SOURCE AND ROOT SECTION
06110 *       NEXT WE READ IN SECTION AT WHICH OUTPUT IS DESIRED.
06120   610 WRITE(42,10027)
06130       READ,KNSECT
06140       WRITE(42,10028)KNSECT
06150       WRITE(42,10001)
06160       READ,ITEST
06170       IF(ITEST.NE.0)GOTO610
06180       IF(KNSECT.GT.9999)GOTO650
06190       IF(INLCSA(KNSECT,1).GE.0)GOTO670
06200       WRITE(42,10029)KNSECT,NIDEN
06210       GOTO610
```

```
06220 *       THIS SECTION HAS BEEN ADDED TO COMPUTE TO ALL TERMINAL NODES
06230 *       ONLY SINGLE LINE OUTPUT IS PROVIDED
06240    600 IALL=1
06250        J=0
06260        DO 630 I=1,NSIZE
06270        IF(INLCSA(I,1).LT.0)GOTO630
06280        IF(INLCSA(I,3).NE.0.OR.INLCSA(I,4).NE.0)GOTO630
06290        J=J+1
06300        INSEQV(J)=I
06310    630 CONTINUE
06320        MEND=J
06330        JQV=J
06340    640 IF(JQV.EQ.0)GOTO870
06350        KNSECT=INSEQV(JQV)
06360        JQV=JQV-1
06370        GOTO670
06380    650 DO 660 I=1,9
06390        LPRINT(I)=0
06400    660 CONTINUE
06410        LPRINT(9)=1
06420        GOTO620
06430 *       MAKE STACK TO PROCESS INFORMATION
06440 *         SINCE THE DESIRED SECTION COULD BE ANYWHERE, THE CURRENT
06450 *         NECO IS UNDEFINED.  IT WILL BE DEFINED WHEN FIRST
06460 *         UPSTREAM SECTION FOUND WITH STATUS DIGIT=2
06470    670 CONTINUE
06480        ISP=1
06490        ND=KNSECT
06500    680 LSDIG=MOD(INLCSA(ND,13),10)
06510 *       NSECTS AT WHICH VOLTAGE-CURRENT HAVE BEEN DETERMINED
06520 *       WILL HAVE STATUS DIGIT=2
06530        IF(LSDIG.EQ.0)GOTO690
06540        IF(LSDIG.GE.2)GOTO700
06550        ISTAK(ISP)=ND
06560        ISP=ISP+1
06570 *       NOW GET A PARENT NSECT,WORKING TOWARDS SOURCE, WHICH
06580 *       HAS BEEN EVALUATED
06590        ND=INLCSA(ND,2)
06600        GOTO680
06610    690 WRITE(42,10030) ND
06620        STOP 4700
06630    700 CONTINUE
06640 *       INLCSA(ND,14)CONTAINS RECORD INDEX
06650        IAREC=INLCSA(ND,14)
06660        LTYP=INLCSA(ND,6)
06670        ASSIGN 710 TO NEXSTA
06680        GOTO550
06690    710 NSECT=KODE
06700        CALL CMTCPY(VTTOTL,VTTTP JRLCU,NECO,NDIM)
06710 *       NOW HAVE READ RECORD ASSOCIATED WITH NSECT WHICH HAS
06720 *       BEEN EVALUATED.  THIS WILL ESTABLISH NECO.
06730 *       IT SHOULD BE NON ZERO
```

```
06740         NECO=NPECO
06750 *       WRITE(LDOUT,10012)'YOUT',NSECT
06760 *       CALL CMTPRT(YOUT,NECO,NDIM,LDOUT)
06770 *       WRITE(LDOUT,10012)'Y INPUT',NSECT
06780 *       CALL CMTPRT(YIN,NECO,NDIM,LDOUT)
06790 *       WRITE(LDOUT,10012)'VTOPH',NSECT
06800 *       CALL CMTPRT(VTOPH,NECO,NDIM,LDOUT)
06810 *         SAVE VPHOUT,YOUT,Y2K IN VOUTMP,YTEMP,YSORPH WHICH ARE
06820 *         EACH OF DIMENSION NECO (IF AT OUTPUT)
06830         XLTEMP=XLENTH
06840         DO 720 I=1,NECO
06850         VOUTMP(I)=VPHOUT(I)
06860         DO 720 J=1,NECO
06870         YTEMP(I,J)=YOUT(I,J)
06880         YSORPH(I,J)=Y2K(I,J)
06890    720  CONTINUE
06900 *       WRITE(42,10031)'ISTAK',(ISTAK(I),I=1,ISP)
06910    730  IF(ISP.EQ.1)GOTO840
06920 *         ISP=1 INDICATES SPECIFIED DESIRED SECTION WHERE RESPONSE
06930 *         IS DESIRED
06940 *         OTHERWISE GET NEXT SECTION IN DOWNSTREAM DIRECTION.
06950 *         NOTE A CURRENT NECO IS DEFINED AT THIS POINT
06960         ISP=ISP-1
06970         ND=ISTAK(ISP)
06980 *         IN THE NEXT SECTION WE OBTAIN SIBLING INFORMATION
06990         DO 740 I=1,NDIM
07000         DO 740 J=1,NDIM
07010    740  YSIB(I,J)=ZERO
07020         NPAR=INLCSA(ND,2)
07030         NSIB=INLCSA(NPAR,3)
07040         IF(NSIB.EQ.ND)NSIB=INLCSA(NPAR,4)
07050         IF(NSIB.EQ.0)GOTO760
07060         IAREC=INLCSA(NSIB,14)
07070         LTYP=INLCSA(NSIB,6)
07080         ASSIGN 750 TO NEXSTA
07090         GOTO550
07100    750  CALL CMTCOP(YIN,YSIB,LCU,LCU,NDIM)
07110    760  IAREC=INLCSA(ND,14)
07120         LTYP=INLCSA(ND,6)
07130         NSECT=ND
07140         ASSIGN 770 TO NEXSTA
07150 *         READ NEXT DOWNSTREAM SECTION DATA
07160 *         YIN,Y2K,VTOPH,APHI,BPHI,CPHI,DPHI,YLOAD,LCU,NECRO
07170 *         REMEMBER AT THIS POINT THE FOLLOWING DATA FROM PARENT HAS
07180 *         BEEN SAVED: VTEMP=YOUTP, YSORPH=Y2KP, VOUTMP=VPHOUTP
07190 *         WHERE P=PARENT
07200         GOTO550
07210 *         NOW WE BEGIN PROCESSING LOOP
07220    770  IF(KODE.NE.NSECT)CALL FILSTO(MDPRL,'NSECT #',NSECT,KODE,3305)
07230 *       WRITE(LDOUT,10012)'Y INPUT',NSECT
07240 *       CALL CMTPRT(YIN,NECO,NDIM,LDOUT)
07250 *       WRITE(LDOUT,10012)'VTOPH',NSECT
```

```
07260 *      CALL CMTPRT(VTOPH,NECO,NDIM,LDOUT)
07270 *      WRITE(LDOUT,10012)'TOTAL LOAD',NSECT
07280 *      CALL CMTPRT(Y2K,NECO,NDIM,LDOUT)
07290 *        NOTE YTEMP,YSORPH,VOUTMP HOLD DATA FROM PARENT NSECT
07300 *        NOTE THE FOLLOWING PROCEDURE COULD BE MADE MORE EFFICIENT BY
07310 *        BY USING A TEMP VARIABLE FOR VOUTMP
07320        XLENTH=XLTEMP+IMLCSA(NSECT,7)/10.0/PMULT
07330        XLTEMP=XLENTH
07340 *      WRITE(LDOUT,10011)'LCU  NECO  NRECO',LCU,NECO,NRECO
07350 *        NOW PERFORM UNIVERSAL ARITHMETIC INDEPENDANT NEARLY OF
07360 *        NREAD
07370 *        LOGIC FOR NRECO,NECO,LCU.  NECO AT THIS POINT REPRESENTS
07380 *        THE NUBER OF EFFECTIVE CONDUCTORS AT OUTPUT OF UPSTREAM
07390 *        SECTION.  LCU IS DETERMINED FROM READ MACRO FOR THE NEW
07400 *        SECTION AND SHOULD BE EQUAL TO NECO.
07410        IF(LCU.NE.NECO)CALL ERRSTT('NTWKAN',0130,NSECT,LCU,NECO)
07420 *        NOW RESET NECO TO NRECO WHICH CAME FROM READ MACRO FOR NEW SECTION
07430        NECO=NRECO
07440 ******************INSER CONSISTANCY CHECKS HERE
07450    772 CONTINUE
07460        DO 790 I=1,NDIM
07470        CUPHI(I)=ZERO
07480        VPHOUT(I)=ZERO
07490        VPHIN(I)=VOUTMP(I)
07500        CNPHI(I)=ZERO
07510        IF(I.GT.LCU)GOTO790
07520        DO 780 J=1,LCU
07530        CUPHI(I)=CUPHI(I)+YIN(I,J)*VOUTMP(J)
07540        YTEMP(I,J)=YTEMP(I,J)+YSORPH(I,J)
07550 *        NOTE THAT THE FOLLOWING NORTON SOURCE PARAMETERS PERTAIN
07560 *        TO THE INPUT TO THE NEW SECTION BEING PROCESSED
07570 *          YTEMP=YOUTP+Y2KP=YOUTP+YINNEW+YINSIB+YLOADP
07580 *                =YNORPH+YINSELF       YINSELF=YIN(I,J)
07590        YNORPH(I,J)=YTEMP(I,J)-YIN(I,J)
07600        CNPHI(I)=CNPHI(I)+YTEMP(I,J)*VOUTMP(J)
07610        IF(I.GT.NECO)GOTO780
07620        VPHOUT(I)=VPHOUT(I)+VTOPH(I,J)*VOUTMP(J)
07630    780 CONTINUE
07640    790 CONTINUE
07650        ASSIGN 730 TO NEXSTA
07660        CALL CMTMPY(VTOPH,VTTMP,VTTOTL,NECO,LCU,JRLCU,NDIM)
07670        IF(NREAD.EQ.1)GOTO800
07680        IF(NREAD.EQ.2)GOTO831
07690 *        ARRIVE HERE FOR 100<=LTYP<=999, NREAD=3
07700 *        IE NORMAL LINE PROCESSING
07710 *      WRITE(LDOUT,10012)'INPUT VOLTAGE',NSECT
07720 *      CALL CMTPRT(VPHIN,NECO,NDIM,LDOUT)
07730 *      WRITE(LDOUT,10012)'OUTPUT VOLTAGE',NSECT
07740 *      CALL CMTPRT(VPHOUT,NECO,NDIM,LDOUT)
07750        CALL CMAPBC(DPHI,YNORPH,BPHI,YSORPH,LCU,LCU,LCU,NDIM)
07760        CALL CMTINV(YSORPH,LCU,LCU,NDIM,WKAREA)
07770        CALL CMAPBC(CPHI,YNORPH,APHI,YTEMP,LCU,LCU,LCU,NDIM)
```

```
07780         CALL CMTMPY(YSORPH,YTEMP,YOUT,LCU,LCU,LCU,NDIM)
07790 *         AFTER POWER CALCULATION, STATUS DIGIT UPDATE AND RECORD
07800 *         WRITE, RETURN TO PROCESS NEXT NSECT
07810         GOTO390
07820     800 CONTINUE
07830 *         ARRIVE HERE FOR 1<=LTYP<=32, NREAD=1
07840 *         IE TRANSISTION AND TRANSPOSITION PROCESSING
07850 *         IN THE FOLLOWING SINCE THE NORTONS ADMITTANCE INVOLVES
07860 *         PARTITIONING WITH THE CURRENTS INTO OTHER TERMINALS ZERO
07870 *         WE FIRST INVERT YNORPH,PARTITION AND THEN RE-INVERT TO GET YOUT
07880         CALL CMTINV(YNORPH,LCU,LCU,NDIM,WKAREA)
07890         CALL CMTMPY(VTOPH,YNORPH,YTEMP,NECO,LCU,LCU,NDIM)
07900 *         NOTE DPHI CONTAINS (FROM NTWKER) VTOPH TRANSPOSE
07910         CALL CMTMPY(YTEMP,DPHI,YOUT,NECO,LCU,NECO,NDIM)
07920         CALL CMTINV(YOUT,NECO,NECO,NDIM,WKAREA)
07930 *         IN THE NEXT STEP YNORPH IS RE-INVERTED  TO RESTORE IT
07940 *         TO ITS ORIGINAL VALUE
07950         CALL CMTINV(YNORPH,LCU,LCU,NDIM,WKAREA)
07960         GOTO390
07970     831 CONTINUE
07980 *         ARRIVE HERE FOR 33<=LTYP<=99, NREAD=2
07990 *         IE RATIO BANK PROCESSING
08000         Y11=YTEMP(1,1)
08010 *         HEREAFTER Y11 BECOMES A TEMPORARY COMPLEX VARIABLE
08020         IF(NECO.EQ.2)GOTO832
08030         IF(NECO.NE.1)STOP 0832
08040 *         NECO=1
08050         Y11=XN*XN*(YRBTR+Y11)/(AONE+(YRBTR+Y11)*ZRBTR)
08060         GOTO833
08070     832 CONTINUE
08080 *         NECO=2
08090         Y22=YTEMP(2,2)
08100         Y12=(YTEMP(1,2)+YTEMP(2,1))/2.
08110         Y11=(Y11+Y12)*(Y22+Y12)/(Y11+Y22+Y12+Y12)
08120         Y11=XN*XN*(YRBTR+Y11-Y12)/(AONE+(Y11-Y12)*ZRBTR)
08130     833 CONTINUE
08140         YOUT(1,1)=Y11
08150         YOUT(2,2)=Y11
08160         YOUT(1,2)=-Y11
08170         YOUT(2,1)=-Y11
08180         GOTO390
08190 *         ARRIVE HERE AFTER DESIRED NSECT IS REACHED AND PROCESSED
08200 *         NEXT TERMINAL PRINT OUT FOR FINAL NSECT.
08210     840 ASSIGN 850 TO NEXSTA
08220         IF(IALL.GT.0)GOTO850
08230         IPRFLG=0
08240         GOTO390
08250     850 IF(NPERT.EQ.1)GOTO910
08260         IF(NPERT.EQ.2)GOTO1060
08270         LDOTMP=LDOUT
08280         IPRFLG=0
08290         LDOUT=42
```

```
08300         ASSIGN 860 TO NEXSTA
08310         GOTO390
08320 *         ABOVE DOES PRINT OUT
08330    860  LDOUT=LDOTNP
08340         IF(IALL.GT.0)IHEAD=1
08350         IF(IALL.GT.0)GOTO640
08360         WRITE(42,10032)
08370         WRITE(42,10001)
08380         READ,ITEST
08390         IF(ITEST.NE.1)GOTO610
08400    870  IALL=0
08410         WRITE(42,10033)
08420         WRITE(42,10001)
08430         READ,ITEST
08440         IF(ITEST.NE.1)GOTO900
08450 *         LET USER CHOOSE ANOTHER CURRENT SOURCE
08460    880  WRITE(42,10034)
08470         WRITE(42,10001)
08480         READ,ITEST
08490         IF(ITEST.NE.1)GOTO190
08500 *         PERTURBATION WILL GO IN HERE
08510         WRITE(LDPRL'14)(INLCSA(NINDEX,0),NINDEX=1,MAXND)
08520         WRITE(42,10035)
08530         WRITE(42,10001)
08540         READ,ITEST
08550         PRINT,'OUTFILE=',MDOUT
08560         IF(IPFLG.EQ.0)GOTO895
08570         ID=INLCSA(NSECT,6)
08580         MV=3
08590         IXC='LENGTH IN METERS'
08600         IYC='VOLTAGE MAGNITUDE'
08610         ITI='PLOT OF VOLTAGE VS. DISTANCE'
08620         X1=0
08630         X2=XLENTH
08640         NXC=16
08650         NYC=17
08660         NTI=28
08670         XY1=XA(1,2)
08680         XY2=XA(1,2)
08690         DO 890 IROW=1,200
08700         DO 890 ICOL=2,4
08710         IF(XA(IROW,ICOL).LT.XY1)XY1=XA(IROW,ICOL)
08720         IF(XA(IROW,ICOL).GT.XY2)XY2=XA(IROW,ICOL)
08730    890  CONTINUE
08740         CALL ZPLOTM(ID,INDPLT,MV,XA,NDIPLT,4,IXC,IYC,
08750        &              ITI,X1,X2,NXC,NYC,NTI,XY1,XY2)
08760    895  CONTINUE
08770         IF (ITEST.NE.0)STOP 5800
08780         CALL DETACH(LDPRL,ISTAT,)
08790         CALL DETACH(LDOUT,ISTAT,)
08800         GOTO20
08810 *         THIS SECTION IS THE IMPLEMENTATION OF PERTURBATION THEORY
```

```
08820 *       AND PERMITS THE DETERMINATION OF EFFECTS DUE TO VARIATION
08830 *       IN SINGLE LOADS AT ANY POINT IN NETWORK
08840 *       IN WHAT FOLLOWS NSECLD IS SECTION WHERE LOAD CHANGE TAKES PLACE
08850 *       NSECRL IS SECTION AT WHOSE END WE DESIRE TO OBTAIN THE RESULT
08860   900 WRITE(42,10036)
08870       READ,NSECLD
08880       WRITE(42,10028)NSECLD
08890       WRITE(42,10001)
08900       READ,ITEST
08910       IF(ITEST.NE.0)GOTO900
08920 *       NEXT WE DETERMINE IF NSECLD HAS BEEN ANALYZED IF NOT WE ANALYZE
08930 *       IT
08940       NPERT=1
08950       ND=NSECLD
08960       LSDIG=MOD(INLCSA(ND,13),10)
08970       KNSECT=ND
08980       IF(LSDIG.LT.2)GOTO670
08990 *       UPON RETURN WE ARE ASSURED HAT NSECLD HAS BEEN ANALYZED
09000 *       WE MUST ONCE AGAIN READ FILE TO ASCERTAIN LCU AND NECO
09010   910 LTYP=INLCSA(NSECLD,6)
09020       IAREC=INLCSA(NSECLD,14)
09030       ASSIGN 950 TO NEXSTA
09040       GOTO550
09050   920 WRITE(42,10037)NECO
09060       READ,(CUREV(I),I=1,NECO)
09070       CALL CVEPRT(CUREV,NECO,NDIM,42)
09080       WRITE(42,10001)
09090       READ,ITEST
09100       IF(ITEST.NE.0)GOTO920
09110   930 WRITE(42,10038)NECO*NECO
09120       READ,((DLY2(I,J),J=1,NECO),I=1,NECO)
09130       CALL CMTPRT(DLY2,NECO,NDIM,42)
09140   940 CONTINUE
09150       WRITE(42,10001)
09160       READ,ITEST
09170       IF(ITEST.NE.0)GOTO930
09180       GOTO980
09190   950 NECO=NRECO
09200       NLDRCO=NRECO
09210       DO 960 I=1,NDIM
09220   960 CUREV(I)=ZERO
09230       WRITE(42,10039)
09240       WRITE(42,10001)
09250       READ,ITEST
09260       IF(ITEST.NE.1)GOTO920
09270       WRITE(42,10040)NECO*NECO,NECO,NECO
09280       READ,((DLY2(I,J),J=1,NECO),I=1,NECO)
09290       CALL CMTPRT(DLY2,NECO,NDIM,42)
09300   970 CONTINUE
09310       WRITE(42,10001)
09320       READ,ITEST
09330       IF(ITEST.NE.0)GOTO950
```

```
09340 *        NEXT COMPUTE DELV2,CUPTB AT NSECLD
09350 *        NOTE=THIS DEPENDS ONLY ON DLY2 AND SECTION PARAMETERS
09360 *             IT IS INDEPENDENT OF RESULT SECTION.
09370 *          EQ. 22 PG 6 OF MATRIX ANALYSIS OF EFFECTS REPORT
09380    980 DO 1000 I=1,NECO
09390        DO 990 J=1,NECO
09400        YTEMP2(I,J)=YOUT(I,J)+Y2K(I,J)+DLY2(I,J)
09410    990 CONTINUE
09420   1000 CONTINUE
09430        CALL CMTZRO(YTEMP2,NECO,NECO,NDIM)
09440        CALL CMTINV(YTEMP2,NECO,NECO,NDIM,WKAREA)
09450        CALL CMTMPY(YTEMP2,DLY2,YTEMP,NECO,NECO,NECO,NDIM)
09460        CALL CMTZRO(YTEMP,NECO,NECO,NDIM)
09470        DO 1020 I=1,NDIM
09480        DELV2(I)=ZERO
09490        IF(I.GT.NECO)GOTO1020
09500        DO 1010 J=1,NECO
09510        DELV2(I)=DELV2(I)-YTEMP(I,J)*VPHOUT(J)
09520   1010 CONTINUE
09530   1020 CONTINUE
09540        DO 1040 I=1,NDIM
09550        CUPTB(I)=ZERO
09560        IF(I.GT.NECO)GOTO1040
09570        CUPTB(I)=CUREV(I)
09580        DO 1030 J=1,NECO
09590        CUPTB(I)=CUPTB(I)-DLY2(I,J)*(VPHOUT(J)+DELV2(J))
09600   1030 CONTINUE
09610   1040 CONTINUE
09620 *        IN THE NEXT SECTION THE USER DEFINES   RESULT SECTION
09630 *        AND WE VERIFY THAT THE SECTION HAS B  N ANALYZED
09640 *        IF NOT, WE PROCEED TO ANALYZE IT.
09650   1050 WRITE(42,10041)
09660        READ,NSECRL
09670        WRITE(42,10028)NSECRL
09680        WRITE(42,10001)
09690        READ,ITEST
09700        IF(ITEST.NE.0)GOTO1050
09710        NPERT=2
09720        ND=NSECRL
09730        LSDIG=MOD(INLCSA(ND,13),10)
09740        KNSECT=ND
09750        IF(LSDIG.LT.2)GOTO670
09760 *        WE ONCE AGAIN READ FILE TO ASCERTAIN LCU AND NECO
09770   1060 LTYP=INLCSA(NSECRL,6)
09780        IAREC=INLCSA(NSECRL,14)
09790        ASSIGN 1070 TO NEXSTA
09800        GOTO550
09810 *        UPON RETURN WE ARE ASSURED THAT NSECRL HAS BEEN ANALYZED
09820 *        NEXT WE DETERMINE LOCATION OF NSECRL RELATIVE TO NSECCD
09830   1070 ND=NSECRL
09840        NECO=NRECO
09850        I=0
```

```
09860         IF(NSECRL.NE.NSECLD)GOTO1100
09870         WRITE(42,10043)
09880         CALL CVEPRT(VPHOUT,NECO,NDIM,42,'VOLTAGE BEFORE CHANGE')
09890         DO 1080 I=1,NECO
09900         DELV2(I)=ZERO
09910         DO 1080 J=1,NECO
09920         DELV2(I)=DELV2(I)+YTEMP2(I,J)*CUPTB(J)
09930   1080  CONTINUE
09940         WRITE(42,10044)
09950         CALL CVEPRT(DELV2,NECO,NDIM,42,'CHANGE IN VOLTAGE')
09960         DO 1090 NA=1,NECO
09970         VSUM(NA)=VPHOUT(NA)+DELV2(NA)
09980   1090  CONTINUE
09990         WRITE(42,10045)
10000         CALL CVEPRT(VSUM,NECO,NDIM,42,'VOLTAGE AFTER CHANGE')
10010         GOTO1420
10020   1100  ND=INLCSA(ND,2)
10030         IF(ND.EQ.NSECLD)GOTO1150
10040         IF(ND.EQ.0)GOTO1110
10050         IF(INLCSA(ND,3).EQ.0.OR.INLCSA(ND,4).EQ.0)GOTO1100
10060         I=I+1
10070         ISTAK(I)=ND
10080         GOTO1100
10090   1110  ND=NSECLD
10100   1120  ND=INLCSA(ND,2)
10110         IF(ND.EQ.NSECRL)GOTO1280
10120         IF(ND.NE.0)GOTO1130
10130         WRITE(42,10042)
10140         STOP 1154
10150   1130  IF(INLCSA(ND,3).EQ.0.OR.INLCSA(ND,4).EQ.0)GOTO1120
10160         DO 1140 J=1,I
10170         IF(ND.EQ.ISTAK(J))GOTO1410
10180   1140  CONTINUE
10190         GOTO1120
10200 *         THE FOLLOWING CODE CORRESPONDS TO THE SITUATION WHEN NSECRL
10210 *         IS DOWNSTREAM FROM NSECLD. IN THIS CASE DELV1=T DELV3
10220 *         FIRST WE CREATE DELV3
10230 *         AT THIS POINT NECO IS THAT OF RESULT NODE IT MUST BE RESET
10240 *         TO LOAD CHANGE NODE I.E. TO NLDRCO
10250   1150  NECO=NLDRCO
10260         DO 1170 I=1,NDIM
10270         DELV3(I)=ZERO
10280         IF(I.GT.NECO)GOTO1170
10290         DO 1160 J=1,NECO
10300         DELV3(I)=DELV3(I)+YTEMP2(I,J)*CUPTB(J)
10310   1160  CONTINUE
10320   1170  CONTINUE
10330         NKAY=NSECLD
10340   1180  I=1
10350         ND=NSECRL
10360   1190  ISTAK(I)=ND
10370         ND=INLCSA(ND,2)
```

```
10380         IF(ND.EQ.NKAY)GOTO1200
10390         I=I+1
10400         GOTO1190
10410 *         THIS SEGMENT ANALYZES THE DLTV1 OF EACH SECTION IN STACK
10420  1200   J=I
10430  1210   ND=ISTAK(J)
10440         IAREC=INLCSA(ND,14)
10450         LTYP=INLCSA(ND,6)
10460         ASSIGN 1220 TO NEXSTA
10470         GOTO550
10480 *         UPON RETURN HERE THE VALUE OF LCU READ SHOULD  EQUAL NECO
10490  1220   NECO=NRECO
10500         DO 1240 K=1,NDIM
10510         DELV1(K)=ZERO
10520         IF(K.GT.NECO)GOTO1240
10530         DO 1230 L=1,LCU
10540         DELV1(K)=DELV1(K)+VTOPH(K,L)*DELV3(L)
10550  1230   CONTINUE
10560  1240   CONTINUE
10570         IF(J.EQ.1)GOTO1260
10580         J=J-1
10590         DO 1250 K=1,NECO
10600         DELV3(K)=DELV1(K)
10610  1250   CONTINUE
10620         GOTO1210
10630 *         NOTE LAST RECORD READ IS AT NSECRL AND NECO OF OUTPUT IS
10640 *          ESTABLISHED
10650  1260   WRITE(42,10043)
10660         CALL CVEPRT(VPHOUT,NECO,NDIM,42,'VOLTAGE BEFORE CHANGE')
10670         WRITE(42,10044)
10680         CALL CVEPRT(DELV1,NECO,NDIM,42,'CHANGE IN VOLTAGE')
10690         DO 1270 NA=1,NECO
10700         VSUM(NA)=VPHOUT(NA)+DELV1(NA)
10710  1270   CONTINUE
10720         WRITE(42,10045)
10730         CALL CVEPRT(VSUM,NECO,NDIM,42,'VOLTAGE AFTER CHANGE')
10740         GOTO1420
10750  1280   NJAY=NSECRL
10760         IBRAN=0
10770  1290   ND=NSECLD
10780         I=1
10790  1300   ISTAK(I)=ND
10800         ND=INLCSA(ND,2)
10810         IF(ND.EQ.NJAY)GOTO1310
10820         I=I+1
10830         GOTO1300
10840 *         THIS SECTION DOES THE ANALYSIS
10850  1310   J=I
10860         ND=ISTAK(J)
10870         IAREC=INLCSA(ND,14)
10880         LTYP=INLCSA(ND,6)
10890         ASSIGN 1320 TO NEXSTA
```

```
10900         GOTO550
10910 *         RECORD READ ESTABLISHES LCU AND NRECO FORSECTION AFTER NJAY
10920 *         THE LCU VALUE IS SAVED AS JLCU AND BECOMES DIMENSION OF RESULT
10930   1320  NECO=NRECO
10940         JLCU=LCU
10950         CALL CMTADD(YIN,YNORPH,ZIN,LCU,NDIM)
10960         CALL CMTINV(ZIN,LCU,LCU,NDIM,WKAREA)
10970 *         NOTE VTOPH HAS NECO ROWS AND LCU COLUMNS
10980         CALL CMTRAN(VTOPH,YTEMP,NECO,LCU,NDIM)
10990 *         YTEMP NOW WILL HAVE LCU ROWS AND NECO COLUMNS
11000 *         YTEMP1 WILL ALSO HAVE LCU ROWS AND NECO COLUMNS
11010         CALL CMTMPY(ZIN,YTEMP,YTEMP1,LCU,LCU,NECO,NDIM)
11020         CALL CMTCPY(YTEMP1,ZIN,LCU,NECO,NDIM)
11030 *         FROM NOW ON FIRST DIMENSION WILL BE JLCU
11040   1330  IF(J.EQ.1)GOTO1350
11050         J=J-1
11060         ND=ISTAK(J)
11070         IAREC=INLCSA(ND,14)
11080         LTYP=INLCSA(ND,6)
11090         ASSIGN 1340 TO NEXSTA
11100         GOTO550
11110 *         NOTE WHEN RETURN FROM READ ZIN WILL BE JLCU X NECO
11120 *         NECO OF OLD SECTION WILL BE EQUAL TO LCU OF NEW ONE
11130   1340  NECO=NRECO
11140         CALL CMTRAN(VTOPH,YTEMP,NECO,LCU,NDIM)
11150         CALL CMTMPY(ZIN,YTEMP,YTEMP1,JLCU,LCU,NECO,NDIM)
11160         CALL CMTCPY(YTEMP1,ZIN,JLCU,NECO,NDIM)
11170         GOTO1330
11180 *         IN THE FOLLOWING DELV3 MUST COME OUT WITH JLCU ELEMENTS
11190 *         ALSO LAST SECTION READ WILL BE NSFCLD WHOSE NECO WILL
11200 *         WILL HAVE SAME DIMENSIONS AS CUPTB
11210   1350  DO 1370 K=1,NDIM
11220         DELV3(K)=ZERO
11230         IF(K.GT.JLCU)GOTO1370
11240         DO 1360 L=1,NECO
11250         DELV3(K)=DELV3(K)+ZIN(K,L)*CUPTB(L)
11260   1360  CONTINUE
11270   1370  CONTINUE
11280         IF(IBRAN.EQ.1)GOTO1400
11290         ND=NJAY
11300         IAREC=INLCSA(ND,14)
11310         LTYP=INLCSA(ND,6)
11320         ASSIGN 1380 TO NEXSTA
11330         GOTO550
11340   1380  NECO=NRECO
11350         WRITE(42,10043)
11360         CALL CVEPRT(VPHOUT,NECO,NDIM,42,'VOLTAGE BEFORE CHANGE')
11370         WRITE(42,10044)
11380         CALL CVEPRT(DELV3,NECO,NDIM,42,'CHANGE IN VOLTAGE')
11390         DO 1390 NA=1,NECO
11400         VSUM(NA)=VPHOUT(NA)+DELV3(NA)
11410   1390  CONTINUE
```

```
11420         WRITE(42,10045)
11430         CALL CVEPRT(VSUM,NECO,NDIM,42,'VOLTAGE AFTER CHANGE')
11440         GOTO1420
11450   1400  NKAY=NJAY
11460         GOTO1180
11470   1410  ISRAN=1
11480         NJAY=ND
11490         GOTO1290
11500   1420  WRITE(42,10046)
11510         WRITE(42,10001)
11520         READ,ITEST
11530         IF(ITEST.NE.1)GOTO1050
11540         WRITE(42,10047)
11550         WRITE(42,10001)
11560         READ,ITEST
11570         IF(ITEST.NE.1)GOTO900
11580         NPERT=0
11590         GOTO880
11600 *         NOW DONE EXCEPT RUN PERTURBATION
11610 *         THIS PROGRAM USES FOLLOWING PSUEDO MACRO SEQUENCES
11620 *         40L SEQUENCE
11630 *         DOES POWER CALCULATION
11640 *         SETS STATUS DIGIT TO 2
11650 *         WRITES NT     FILE RECORD
11660 *         RETURNS TO NEXSTA
11670 *         201-204 SEQUENCE
11680 *         READS NT     FILE AND IN ALL CASES GETS LCU AND NRECO
11690 *         IF NECO IS UNDEFINED (NECO=-1) IT SETS NECO=NRECO
11700 *         RETURNS TO NEXSTA
11710 *         211-214 SEQUENCE
11720 *         WRITES NT     RECORD
11730 *         GOTO105 SEQUENCE
11740 *         105 SEQUENCE
11750 *         WRITES TO ASCII OUTPUT FILE
11760 *         RETURNS TO NEXSTA
11770 *    CNPHI      NORTON EQUIVALENT SOURCE CURRENT AT OUTPUT OF PARENT
11780 *               SECTION-PHASE
11790 *    CSPHI      SOURCE CURRENT VECTOR-PHASE
11800 *    CUPHI      CURRENT VECTOR INTO SECTION-PHASE
11810 *    DAPRYLIJ   RANDOM BINARY FILE CONTAINING PRIMARY LOAD DATA
11820 *    DARBTRLI   RANDOM BINARY FILE CONTAINING RATIO BANK TRANSFORMER DATA
11830 *    DASECDLI   RANDOM BINARY FILE CONTAINING SECONDARY LOAD DATA
11840 *    DATAIN     ASCII INPUT FILE
11850 *    DATRANLI   RANDOM BINARY FILE CONTANING DISTRIBUTION TRANSFORMER DATA
11860 *    DAWKINIJ   ASCII FILE CONTAINING NETWORK LOGICAL DESCRIPTON
11870 *    DLTVO      COMPLEX MATRIX
11880 *    DLY2       COMPLEX MATRIX REPRESENTING OFF DIAGONAL ELEMENTS OF Y2LEX
11890 *    DLYIN      COMPLEX MATRIX APPROXIMATION TO YINEX
11900 *    IAREC      RANDOM RECORD ADDRESS ASSOCIATED WITH FILE DARBTRLI
11910 *    IASN       ASSUMPTION CODE
11920 *    ICOND      INTEGER LENGTH CODE FROM DNWKINLI
11930 *    IDATIN     TERMINAL (EQ.0) VS FILE (NE.0) CONTROL FLAG FOR INPUT DATA
```

```
11940 *    IDRBTR    RATIO BANK TRANSFORMER IDENT READ FROM DARBTRLI
11950 *    IPRFLG    FLAG USED TO CONTROL PRINTOUT
11960 *    IELEM
11970 *    IFCODE    FREQUENCY CODE
11980 *    IFILE     FLAG TO CONTROL OPENING OF RANDOM FILES TEMPRAR1
11990 *              AND TEMPRAR2
12000 *    IHEAD     HEADER PRINT CONTROL FLAG
12010 *    INDRHO    OHM CODE
12020 *    INSECTS   NUMBER OF ELEMENTS USED (IN CASCADE) TO FORM NETWORK
12030 *    IOMCD     OHMCODE ASSOCIATED WITH CURRENT OPEN DPULIJKN FILE
12040 *    IOPT      OPTION CONTROL OF EXACT, DIAGONAL APPROXIMATE CALCULATIONS
12050 *    IPHS      PHASE CONNECTION CODE
12060 *    IPLTF     PLOT FLAG USED TO PREVENT OVERWRITING PLOT ARRAY
12070 *    IPFLG     PLOT FLAG TO BYPASS PLOTTING LOGIC
12080 *    IPRIM     PRIMARY LOAD CODE
12090 *    IPROC     VECTOR CONTAINING ELEMENT SEQUENCING
12100 *    IQUANT    MODAL (EQ.0) VS PHASE (NE.0) CONTROL FLAG FOR INPUT
12110 *              ADMITTANCES
12120 *    IRECD     RANDOM RECORD ADDRESS ASSOCIATED WITH FILE NTLIJKNM
12130 *    IRES      OHM CODE FROM DNWKINLI
12140 *    IRTRAN    DISTRIBUTION TRANSFORMER TYPE CODE PLUS 1
12150 *    ISECD     SECONDARY LOAD TYPE CODE
12160 *    ISELEC    RANDOM (EQ.0) VS MANUAL (NE.0) DETERMINATION OF SEQUENCE
12170 *    ISTM      ISTM+1=LAST RECORD IN DASECDLI FILE
12180 *    ITEMP     TEMPERATURE CODE
12190 *    ITTM      ITTM+1=LAST RECORD IN DATRANLI FILE
12200 *    ITTY      DISTRIBUTION TRANSFORMER TYPE CODE
12210 *    IVOLT     CURRENT (ED.0) VS VOLT (NE.0) CONTROL FLAG FOR SOURCE
12220 *    IZNGR     INTEGER GROUNDING ADMITTANCE
12230 *    JROOT     ROOT SECTION NUMBER
12240 *    KDRBTR    FLAG TO CONTROL OPENING DARBTRLI
12250 *    KELEM     VECTOR OF FLAGS TO INDICATE IF A SECTION IS VALID
12260 *    KISRC
12270 *    KPRIM     FLAG TO CONTROL OPENING DAPRYLIJ
12280 *    KTRSEC    FLAG TO CONTROL OPENING DASECDLI AND DATRANLI
12290 *    KVTSU     FLAG TO CONTROL OPENING SULIJKNM
12300 *    KYSRC
12310 *    LCU       NUMBER OF EFFECTIVE CONDUCTORS ASSOCIATED WITH
12320 *              NEW YIN (NTWKER)
12330 *    LCU       NUMBER OF EFFECTIVE CONDUCTORS INPUT TO NEW SECTION (NTWKAN)
12340 *    LDAIN=10  FRN FOR DNWKINIJ
12350 *    LDOUT     FRN FOR ASCII OUTPUT FILE
12360 *    LDPRL=20  FRN FOR NTLIJKNM
12370 *    LDPU =25  FRN FOR DPULIJKN PER UNIT LENGTH DATABASE FILE
12380 *    LDSEC=11  FRN FOR DASECDLI SECONDARY LOAD DATABASE FILE
12390 *    LDT1      FRN FOR TEMPRAR1
12400 *    LENGTH    CODED LENGTH
12410 *    LEVND     LEVEL OF SECTION=INLCSA(NSECT,5)
12420 *    LEVPAR    LEVEL OF PARENT
12430 *    LOOTMP    TEMPORARY FRN SAVE
12440 *    LOUT =16  FRN FOR NTWKOUT1
12450 *    LPRIM=12  FRN FOR DAPRYLIK
```

```
12460 *    LRBTR=14  FRN FOR DARBTRLI RATIO BANK TRANSFORMER DATABASE FILE
12470 *    LSDIG     LARGEST DIGIT
12480 *    LSUBN     FRN FOR ASCII SUBNETWORK INPUT PARAMTER FILE SUBNTLMN
12490 *    LTOP      LTYP ASSOCIATED WITH PREVIOUS READ OF DPULIJKN
12500 *    LTRAN=13  FRN FOR DATRANLI DISTRIBUTION TRANSFORMER DATABASE FILE
12510 *    LTYO      DUMMY TO BE DELTED FROM FINAL OPERATIONAL FORM
12520 *    LTYP      LINE TYPE CODE
12530 *    LVTSU     FRN FOR SULIJKNM
12540 *    MAXND=512 DIMENSION LIMIT TO MAXIMUM NUMBER OF SECTIONS
12550 *    MAXRUN
12560 *    MDOUT     CHARACTER VARIABLE FOR ASCII OUTPUT FILE
12570 *    MDPRL     FILENAME VARIABLE FOR NTLIJKNM
12580 *    MDPU      CHARACTER VARIABLE FOR FILENAME DPULIJKN
12590 *    MDT1      CHARACTER VARIABLE='TEMPRAR1'
12600 *    MDT2      CHARACTER VARIABLE='TEMPRAR2'
12610 *    MECO      SAVED VALUE OF NECO FROM HEADER RECOD
12620 *    MNTWK     FILENAME VARIABLE FOR DNWKINIJ
12630 *    MPRIM     CHARACTER VARIABLE FOR FILENAME DAPRYLIK
12640 *    MPRS      FILENAME VARIABLE FOR NTWKOUT1
12650 *    MRBTR     CHARACTER VARIABLE FOR FILENAME DARBTRLI
12660 *    MREC=750  RECORD SIZE FOR NTLIJKNM
12670 *    MTRAN     CHARACTER VARIABLE FOR FILENAME DATRANLI
12680 *    NCHECK    STOP CODE
12690 *    NDIM =4   DIMENSION OF ADMITTANCE ARRAYS
12700 *    NDIPLT    DIMENSION OF PLOTTING ARRARY
12710 *    NDPUZ     RECORD SIZE FOR DPULIJKN
12720 *    NDSEC     CHARACTER VARIABLE FOR FILENAME DASECDLI
12730 *    NECO      NUMBER OF EFFECTIVE CONDUCTORS ASSOCIATED WITH
12740 *              OLD YIN (NTWKER)
12750 *    NECO      NUMBER OF EFFECTIVE CONDUCTORS AT OUTPUT OF PRESENT
12760 *              SECTION (NTWKAN)
12770 *    NIDEN     NETWORK IDENTIFICATION NUMBER
12780 *    NLCSA=14  NUMBER OF RECORDS USED TO CONTAIN INLCSA ARRAY
12790 *    NOMAX     TOTAL NUMBER OF SECTIONS
12800 *    NPAR      PARENT SECTION=INLCSA(NSECT,2)
12810 *    NPER      FLAG USED DURING PERTURBATION ANALYSIS
12820 *    NPRISZ=38 RECORD SIZE FOR DAPRYLIJ
12830 *    NRBTR=78  RECORD SIZE FOR DARBTRLI
12840 *    NRDAT=1   NUMBER OF RECORDS USED TO CONTAIN A SINGLE SECTION DATA
12850 *    NREAD     POINTER TO CORRECT READ FORMAT AND BASED ON LTYP
12860 *    NRECO     NUMBER OF EFFECTIVE CONDUCTORS READ IN FROM
12870 *              DPULIJKN (NTWKER)
12880 *    NRECO     NUMBER OF EFFECTIVE CONDUCTORS AT OUTPUT OF NEW
12890 *              SECTION (NTWKAN)
12900 *    NSECM     FILE DASECDLI CONTROL PARAMETER
12910 *    NSEQV=1   NUMBER OF RECORDS USED TO CONTAIN INSEQV VECTOR
12920 *    NTRNSZ    RECORD SIZE FOR DATRANLI
12930 *    NTRPH     PHASE CONNECTION CODE FOR DISTRIBUTION TRANSFORMER
12940 *    NTWKNO    NETWORK NUMBER (IN CASCADE)
12950 *    NVTSZ     RECORD SIZE FOR SULIJKNM
12960 *    POU
12970 *    POWER
```

```
12980 *    SULIJKNM  RANDOM BINARY FILE CONTAINING COMPUTED SUBNET PARAMETERS
12990 *    TVOD      COMPLEX VECTOR ERPRESENTING DIAGONAL VOLTAGE TRANSFER RATIO
13000 *    TVOEX     COMPLEX MATRIX
13010 *    TVOEX     COMPLEX MATRIX MODAL VOLTAGE TRANSFER RATIO
13020 *    VOUTMP      TEMPORARY STORAGE OF VOLTAGE VECTOR FROM PARENT TO SUN
13030 *    VPHIN       VOLTAGE VECTOR AT INPUT TO SECTION
13040 *    VPHOUT      VOLTAGE VECTOR AT OUTPUT OF SECTION-PHASE
13050 *    VTOPH     VOLTAGE TRANSFER RATIO MATRIX
13060 *    Y2K       SECTION TOTAL TERMINATION ADMITTANCE MATRIX-PHASE
13070 *    Y2LDI     COMPLEX VECTOR REPRESENTING DIAGONAL ELEMENTS OF Y2LEX
13080 *    Y2LEX     COMPLEX MATRIX EXACT SECTION TERMINATION ADMITTANCE
13090 *    YIN       SECTION INPUT ADMITTANCE MATRIX-PHASE
13100 *    YIND      COMPLEX VECTOR REPRESENTING DIAGONAL APPROXIMATION TO YINEX
13110 *    YINEX     COMPLEX MATRIX EXACT SECTION INPUT ADMITTANCE
13120 *    YLOAD     TOTAL EXTERNAL CONNECTED LOAD ADMITTANCE MATRIX-PHASE
13130 *    YLTOT     COMPLEX MATRIX TOTAL SUMMED LOAD ADMITTANCE
13140 *    YNORPH    NORTON EQUIVALENT SOURCE ADMITTANCE AT OUTPUT OF
13150 *              PARENT SECTION-PHASE
13160 *    YOUT      ADMITTANCE MATRIX LOOKING BACK INTO SECTION OUTPUT
13170 *              TERMINALS-PHASE
13180 *    YPUL      LINE ADMITTANCE PER UNIT LENGTH
13190 *    YSORPH    SOURCE ADMITTANCE MATRIX-PHASE (SOMETIMES TEMPORARY ARRAY)
13200 *    ZPUL      LINE IMPEDANCE PER UNIT LENGTH
13210        END
```

```
**************************************************************************

NTWKERS4                                                  06/01/81    9:16 AM

**************************************************************************


00010*# RUNH *=NTWKERH4(ULIB,CORE=46K)LIBRARY4;LIBRARY/IMSL,R;
00020*#                                  LIBRARY/OLDTSLIB,R
00030 *        THIS RCR VERSION CONTAINS MODIFICATIONS FOR:
00040 *           RATIO BANK TRANSFORMER PROCESSING-SINGLE PHASE
00050 *           ELIMINATED MANY UNUSED VARIABLES
00060 *           TIDIED UP STRUCTURE SOMEWHAT
00070 *          READ DPUIJKLM FILES USING A DIRECTORY VECTOR
00080 *          READ DPUIJKLM FILES HAVE ONE RECORD PER LINE TYPE
00090 *          CONSIDERABLE SOPHISTICATED LOGIC FOR CHECKING
00100 *            BINARY TREE CONSISTANCY
00110 *            NECO-LCU-NCU CONSISTANCY
00120 *          LINE SEQUENCING IN DNWKINIJ EXPECTED TO BE 1000,1
00130 *          ALSO THE METHOD OF ADDRESSING LDPRL FILE HAS BEEN
00140 *           CHANGED TO BE NLINE-982 WHICH ACCOUNTS FOR 18
00150 *           HEADER AND INLCSA RECORDS
00160 *         TO OETAIN A COMPLETE DIAGNOSTIC ASCII OUTPUT OF ALL
00170 *         ADMITTANCES WITH ANNOTATION, REMOVE ALL *** COMMENT
00180 *         CHARACTERS-RCRUSTAY
00190 *        IN THIS VERSION JTG HAS ADDED THE FOLLOWING CAPABILITIES:
00200 *          INCREASING LENGTHS BY MULTIPLYING WITH PMULT
00210 *          ADDITION OF REAL LOAD AT EVERY DT FOR DISTRIBUTED LOADING
00220 *          LOADING.  SEE IOPC CONTROL FLAG IMBEDDED IN PROGRAM
00230 *        ALSO AN OPTION HAS BEEN INCLUDED FOR READING DISTRIBUTION
00240 *          TRANSFORMER ADMITTANCE DATA FROM A FILE TRANFILE (EFFECTIVELY)
00250 *          VIA A SUBROUTINE TRANAD.
00260 *
00270 *        THIS PROGRAM DETERMINES THE DRIVING POINT ADMITTANCE
00280 *        MATRICES AND
00290 *        THE VOLTAGE TRANSFER RATIO MATRICES FOR A MULTICONDUCTOR TRANS-
00300 *        MISSION LINE NETWORK. THE NETWORK IS SPECIFIED BY THE USE OF
00310 *        SEVERAL DATA BASES.  THESE ARE:
00320 *          1 DNWKIN12    NETWORK CONNECTION INFORMATION SEQUENCIAL  ASCII
00330 *          2 DPU12345    PER UNIT LENGTH LINE PARAMETERS  RANDOM BINARY
00340 *          3 DASECD12    SECONDARY LOAD ADMITTANCES     RANDOM BINARY
00350 *          4 DRBTR123    ABCD MATRICES FOR RATIO BANK TRANSFORMERS
00360 *                        RANDOM BINARY
00370 *          6 DPRIM123    PRIMARY LOAD DATA FOR LTYPE > 10000  RANDOM BINARY
00380 *          7 DTRANN12    ANALYTIC TRANSFORMER PARAMETER FILE FOR IOPC=0,1,2
00390 *        NOTE=IN ABOVE NUMBERS REPRESENT CONCATENATED STRINGS OF
00400 *          CODE DESIGNATIONS DESIGNED,BY MR. R.C. RUSTAY,TO ALLOW
00410 *          CATALOGING AND RETRIEVAL OF THE APPROPRIATE INFORMATION.
00420 *        REPLACING ALL B***B WITH BBBBB WILL CREATE OUTPUT ASCII FILE
00430 *          WITH COMPLETE ADMITTANCE ANALYSIS.  BESURE TO USE A TEMPORARY
00440 *          ASCII FILE TO SAVE NEW SOURCE. IE DONOT CHANGE ORIGINAL
00450 *        THE OUTPUT FROM THIS PROGRAM WILL CONSIST OF A RANDOM BINARY FILE
00460 *          NT123456 WHICH IS ALSO CODED WITH A CONCATENATED STRING OF CODE
00470 *          DESIGNATIONS.
00480 *        NOTE NDIM=4 LIMITS PROGRAM TO IASN=2,4
00490 *        NOTE THAT ARRAYS AM,BM,CM ARE NOT USED IN THIS  OR COMPANION
```

```
00500 *       PROGRAM NTWKANSI; ARE INCLUDED IN THE FUTURE EVENT THEY MAY BE
00510 *       NEEDED TO REPRESENT OTHER APPARATUS
00520 10000 FORMAT('THIS PROGRAM EVALUATES THE DRIVING POINT ADMITTANCE'/
00530      & 'MATRICES AND VOLTAGE TRANSFER RATIO MATRICES'/'THE FOLLOWING',
00540      & ' SPECIFICATIONS DEFINE THE RUN PARAMETERS :'/'FREQ CODE ',
00550      & ' - 2 DIGITS  ,TEMP CODE  - 1 DIGIT  ',','ASSUM CODE',
00560      & ' - 1 DIGIT  ,NETWORK ID NUMBER - 2 DIGITS'/)
00570 10001 FORMAT(//'FREQUENCY CODE=',I2,10X,'TEMPERATURE CODE=' ,I3/
00580      & 'ASSUMPTION CODE=',I3,9X,'NETWORK NUMBER=' ,I4///Y10X,
00590      & 'IS THIS DATA CORRECT ?'/'IF YES HIT RETURN',5X,'IF NO ENTER 1 ',
00600      & 'THEN HIT RETURN '/)
00610 10002 FORMAT('ENTER PERCENTAGE MAGNITUDE MEAN, STD. DEV AND '/
00620      & ' PHASE ANGLE MEAN,STD.DEV(DEGREES) FOR LOAD')
00630 10003 FORMAT(' MEAN  ',F6.1,' PER CENT ',' MAGNITUDE ',' DEV ',
00640      & F6.1,' PER CENT '/' MEAN ',F6.1,' DEGREES ANGLE  DEV  ',
00650      & F6.1,' DEGREES'/' ENTER 1 IF WRONG ELSE CR')
00660 10004 FORMAT(' MULTIPLYING FACTOR FOR LENGTH=',F10.7)
00670 10005 FORMAT(6HDNWKIN,I2,1H;)
00680 10006 FORMAT(//'NETWORK IDENTIFICATION NUMBER IS IN ERROR'//
00690      & 'ASSIGNED NUMBER=',I4,10X,'DATA NUMBER=',I4//'CHOOSE OPTIONS'/
00700      & ' TYPE 1 TO CONTINUE PROCESSING'/'TYPE 2 TO RETURN TO THE',
00710      & ' BEGINNING OF THE PROGRAM'/'TYPE 3 TO ABORT'/'HIT RETURN')
00720 10007 FORMAT(//'DNWKIN DATA BASE RECORD/READING ERROR ENCOUNTERED' /
00730      & 'RECORD NUMBER ',I4,'  HAS LINE NUMBER ',I4/'PROGRAM ABORTS')
00740 10008 FORMAT(/'ENTRY ON LINE NUMBER',I5,'  IS A SECOND ENTRY FOR',
00750      & ' SECTION ',I5/'PROGRAM ABORTS')
00760 10009 FORMAT(//'ENTRY AT LINE  ',I5,' INDICATES MORE THAN TWO',
00770      & ' BRANCES FROM SECTION',I5/'RECONFIGURE NETWORK AS A BINARY',
00780      & ' TREE BY INSERTING ZERO LENGTH SECTIONS'/'PROGRAM ABORTS')
00790 10010 FORMAT(//'THE DNWKIN FILE CONTAINS TWO ROOT SECTIONS',
00800      & 'THE SECOND ONE IS ON LINE',I5,' AT SECTION ',I5/
00810      & 'THE FIRST ROOT IS ',I5/'PROGRAM ABORTS')
00820 10011 FORMAT(//'ENTRY ON LINE ',I5,' INDICATES DUPLICATE ENTRY OF ',
00830      & 'A SON AT SECTION',I5/'EITHER A LOOP EXISTS OR ENTRY ERROR'/
00840      & 'NETWORK MUST BE CONFIGURED AS A TREE'/'PROGRAM ABORTS')
00850 10012 FORMAT(' AT LINE',I6,'SECTION',I6,' HAS NONEXISTANT SON',I6)
00860 10013 FORMAT('SECTION ',I5,'AT LINE ',I5,' HAS A PARENT SECTION '/
00870      & ' NUMBER ',I5,' AT LINE ',I5, 'WHICH IS A TERMINAL SECTION '/)
00880 10014 FORMAT(' SECTION ',I5,' AT LINE ',I5,' AND ITS PA&RENTY'/
00890      & ' SECTION ',I5,' AT LINE ',I5,' ARE NOT COMPATIBLE '/)
00900 10015 FORMAT('PATH STARTING WITH TERMINAL SECTION ',I5,
00910      & ' IS A LOOP ENCOUNTERING *: '/(15I6)/)
00920 10016 FORMAT(' THE NETWORK APPEARS DISJOINT '/' THE FOLLOWING '
00930      & ,'NON TERMINAL SECTIONS ARE NOT ENCOUNTERED IN PATHS :'/
00940      & (I5 ,' AT LINE ',I5/)/)
00950 10017 FORMAT(' THE FOLLOWING SECTIONS WERE FOUND TO BE IN ERROR :'/
00960      & (10X,I5,' AT LINE NUMBER ',I6)/)
00970 10018 FORMAT(2HNT,I2,2I1,I2,1H;)
00980 10019 FORMAT('WKEROU',I2,';')
00990 10020 FORMAT(A50/(15I5))
01000 10021 FORMAT(A50,I5/8(2X,1PE12.6))
01010 10022 FORMAT(A50,7I6)
```

```
01020 10023 FORMAT(//'OUTPUT FILE NOT AVAILABLE ON ',I6/
01030      & 'TO WRITE RECORD NUMBER ',I6/'PROGRAM ABORTS')
01040 10024 FORMAT(//'OUTPUT FILE WRITE ERROR ON',I6,'FILE AT RECORD ',I6/
01050      & 'PROGRAM ABORTS')
01060 10025 FORMAT(//'PROGRAM ERROR.  INVALID INFORMATION OF SECTION ',I6/
01070      & 'PROGRAM ABORTS')
01080 10026 FORMAT(14I5,I8)
01090 10027 FORMAT('FREQ=',I2,' TEMP=',I1,' ASSUMPTION=',I1,' NETWORK=',I2)
01100 10028 FORMAT(V)
01110 10029 FORMAT(//'FILE CLOSING NOT SUCCESSFUL'/'DO YOU WISH TO CONTINUE?'/
01120      & /'IF YES ENTER 1 AND HIT RETURN ,IF NO HIT RETURN'//)
01130 10030 FORMAT(//'USER INSTRUCT PROGRAM ABORT')
01140 10031 FORMAT(//'PROGRAM WILL CONTINUE WITH FILE IN AFT'/'IF THE AFT',
01150      & 'GETS FULL THE PROGRAM MAY ABORT')
01160 10032 FORMAT(//'FILE',A10,' IS NOT AVAILABLE ON UNIT ',I5/
01170      & 'DUE TO ISTAT CONDITION ',I5/'PROGPAM ABORTS',
01180      & ', CHECK OHMCODE IN FILE',A9)
01190 10033 FORMAT('IOPC,DB,ANGLE=',I2,2F8.2)
01200 10034 FORMAT('SECTION,ITTY,LTYP,LENGTH=',3I6,F8.1)
01210 10035 FORMAT(8I5)
01220 10036 FORMAT('SUBNETWORK AT SECTION=',I6)
01230 10037 FORMAT('1 PHASE TO 3 PHASE AT SECTION=',I6)
01240 10038 FORMAT(/'LOAD INFORMATION AT SECTION ',I5,/ ' INCORRECT'/
01250      & 'PROGRAM ABORTS')
01260 10039 FORMAT('3 PHASE TRANSPOSITION AT SECTION=',I6)
01270 10040 FORMAT(//'THREE PHASE TRANSPOSITION LINE TYPE IN ERROR AT SECTION'
01280      & ,I5/'PROGRAM ABORTS')
01290 10041 FORMAT('RATIO BANK TRANSOFRMER AT SECTION=',I6)
01300 10042 FORMAT(5HDRBTR,I2,I1,1H;)
01310 10043 FORMAT('DIAGNOSTIC DATA FOR READ OF DARBTRLI'/6I7)
01320 10044 FORMAT('LINE TYPE',I4,' AT SECTION',I4)
01330 10045 FORMAT(3HDPU,I2,3I1,1H;)
01340 10046 FORMAT(10(2I4,2X))
01350 10047 FORMAT(/'DIMENSION SIZE OF DATA IS',I5,' WHILE DIMENSION OF ',
01360      & ' MATRIX IS ' I5/'ON LINE TYPE ',I5,' ON FILE ',A10/
01370      & ' PROGRAM ABORTS')
01380 10048 FORMAT(5HDAPRY,I2,I1,1H;)
01390 10049 FORMAT(6HDTRANN,I2,1H;)
01400 10050 FORMAT(/'PROCESSING ERROR AT SECTION ',I5,' ON PASS ',I5/
01410      & 'PROGRAM ABORTS ')
01420 10051 FORMAT(/'LEVEL OF PARENT SECTION ',I5,' IS',I5 /
01430      & ' WHILE THAT OF PRESENT SECTION ',I5,' IS',I5/
01440      & ' PROGRAM ABORTS ')
01450 10052 FORMAT('SPECIAL TRANSFORMER FILENAME=',A8,' FOR IOPC=3')
01460 10053 FORMAT('TYPE CR IF OK')
01470 10054 FORMAT('WCODE,NSECT,IASN,IZNGR =',4I6)
01480 10055 FORMAT('WCODE,NSECT,LTYP =',3I6)
01490 10056 FORMAT('WCODE,NSECT,LTYP,NECO,NEUF =',5I6)
01500 10057 FORMAT('WCODE,NSECT,LTYP,NEUF,NEUFO =',5I6)
01510 10058 FORMAT('WCODE,NSECT,LTYP,IPRI,IP =',5I6)
01520 10059 FORMAT('WCODE,NSECT,LTYP,ITTY,NEUF,NUMF =',6I6)
01530 10060 FORMAT('WCODE,NSECT,LTYP,ITTY,ITREC =',5I6)
```

```
01540 10061 FORMAT('WCODE,NSECT,LTYP,LCU,NECO,NCU =',6I6)
01550 10062 FORMAT('WCODE,NSECT,LTYP,NEUF,NUMF,IPRIM =',6I6)
01560 10063 FORMAT('WCODE,NSECT,LTYP,LCU,NECO,NCU,IPRIM =',7I7)
01570 10064 FORMAT('UNABLE TO FIND LTYP=',I3,' IN FILE=',A9,
01580      &        ' FOR NSECT=',I5)
01590 10065 FORMAT('TRANSFORMER ADMITTANCE MODIFIED BY',F6.1,
01600      &        ' DB AND ANGLE',F7.1,' IN DEGREES')
01610 10066 FORMAT('TRANSFORMER ADMITTANCE MULTIPLIED BY ',1P2E14.4)
01620 10067 FORMAT(1P6E16.6)
01630 10068 FORMAT(A8,';')
01640       IMPLICIT COMPLEX(A-G,Q,S,T,V,Y,Z)
01650       DIMENSION INLCSA(1800,14),INSEQV(1800),ISTAK(1800),ICBUF(5),
01660      & YIN(4,4),Y2K(4,4),YLOAD(4,4),YTEMP(4,4),AM(4),BM(4),IDIR(900),
01670      & CM(4),DM(4),ZNULL(4,4),VNULL(4),LEVPT(50),
01680      & NECOV(50),APHI(4,4),BPHI(4,4),CPHI(4,4),DPHI(4,4),VTOPH(4,4),
01690      & WKAREA(200),KXFER(3),
01700      & Z(4,4),Y(4,4),ZO(4,4),YO(4,4),S(4,4),SI(4,4),
01710      & YPRIM(4,4),ZT(4),YDIAG1(4),YDIAG2(4),
01720      & YSTORE(4,4,50),DL(4),TRANAY(10,9)
01730       CHARACTER *9  WNTWK,MDPRL,MDSEC,MPRIM,MABCD,MDPU,MIDENT,
01740      & MVTSU,MPRS,MRBTR,MTRAN,TRANF
01750       CHARACTER*8 NDATE,NDTE,TTRAN
01760       CHARACTER*6 CN
01770       DATA LDAIN/10/MAXND/1800/LDPRL/20/LDSEC/11/LPRIM/12/LRBTR/14/
01780       DATA LDPU/25/LSUBN/18/LOUT/16/,KOUT/1000/LTRAN/15/ITRAN/0/
01790       DATA NATRSZ/20/,LDTRA/13/,MRBTR/'RBTRDATA;'/
01800       DATA MREC/1800/NRDAT/1/NSEQV/1/NLCSA/14/
01810       DATA NTRNSZ/15/NRBTR/78/NVTSZ/78/NPRISZ/38/
01820       DATA NSECSZ/9/NDPUSZ/210/
01830       DATA ZERO/(0.0,0.0)/AONE/(1.0,0.0)/CN/'NTWKER'/,TTRAN/'TRANFILE'/
01840       DATA KDPRI/0/KDRBTR/0/KTRSEC/0/KVTSU/0/
01850       DATA NDIM/4/LU/06/NWKA/200/,IOPC/3/
01860       DATA IRAN/0/,IDIA/1/,ZNULL/16*(0.,0.)/,VNULL/4*(0.,0.)/
01870       DATA NECO/-1/,LTOP/-1/
01880       DATA XTPI/6.2831851/RAD/57.295779/XDB/0.0/XANG/0.0/
01890 *       THIS SEGMENT OF THE PROGRAM CONSISTS OF INTERACTIVE INPUT.
01900 *       THE USER SPECIFIED CODES WHICH ARE USED TO DETERMINE THE
01910 *       NETWORK TO BE ANALYZED AND ITS PARAMETERS.
01920       CALL DATIM(NDATE,WTIME)
01930       CALL FPARAM(1,120)
01940       CALL NASTRK
01950       PMAGMN=100.0
01960       PMAGST=0.0
01970       PANGMN=0.0
01980       PANGST=0.0
01990       IASN=4
02000       ITEMP=6
02010       PMULT=1.0
02020       LODFLG=0
02030   100 CONTINUE
02040   110 WRITE(06,10000)
02050   120 PRINT,'ITEMP,IASN=',ITEMP,IASN
```

```
02060         WRITE(6,10053)
02070         READ,ITEST
02080         IF(ITEST.EQ.0)GOTO 130
02090         PRINT,'ENTER ITEMP,IASN '
02100         READ,ITEMP,IASN
02110         GOTO 120
02120    130  PRINT,'ENTER FREQUENCY CODE,NETWORK IDENTIFICATION NUMBER'
02130         READ,IFCODE,NIDEN
02140         PRINT,'FREQ CODE=',IFCODE,'  NETWORK ID=',NIDEN
02150         WRITE(6,10053)
02160         READ,ITEST
02170         IF(ITEST.NE.0)GOTO 130
02180    140  WRITE(06,10004)PMULT
02190         WRITE(06,10053)
02200         READ,ITEST
02210         IF(ITEST.EQ.0)GOTO150
02220         READ,PMULT
02230         GOTO140
02240 *         BASE DNWKIN12(12 ARE LOCATIONS FOR NETWORK IDENTIFICATION
02250 *         NUMBER). IT THEN CHECKS TO INSURE CONSISTENCY OF INFORMATION
02260    150  ENCODE(MNTWK,10005)NIDEN
02270         CALL OPENF(LDAIN,MNTWK,ISTAT,1,0,1)
02280         READ(LDAIN,10028)NLINE,KIDEN
02290         IF(NIDEN.EQ.KIDEN)GOTO170
02300         WRITE(LU,10006)NIDEN,KIDEN
02310         READ,NOPTN
02320         GOTO(170,100,160),NOPTN
02330    160  STOP 0100
02340    170  DO 180 I=1,MAXND
02350         INSEQV(I)=0
02360         INLCSA(I,1)=-10
02370    180  ISTAK(I)=0
02380         JROOT=0
02390         NOMAX=0
02400         I=1000
02410         IABORT=0
02420    190  I=I+1
02430         READ(LDAIN,10028,END=260)NLINE,INSECT,IPAR,ILSON,IRSON,
02440        &  LTYP,ILENCD,IRES,ITT,IPHSE,ISECD,IPRIM,IZNGR
02450         NOMAX=NOMAX+1
02460         IF(NLINE.EQ.I)GOTO200
02470         WRITE(06,10007)I,NLINE
02480         IABORT=1
02490         GOTO190
02500    200  IF(INLCSA(INSECT,1).LT.0)GOTO210
02510         WRITE(06,10008)NLINE,INSECT
02520         IABORT=1
02530         GOTO190
02540    210  INLCSA(INSECT,1)=INSECT
02550         INLCSA(INSECT,2)=IPAR
02560         INLCSA(INSECT,3)=ILSON
02570         INLCSA(INSECT,4)=IRSON
```

```
02580         INLCSA(INSECT,6)=LTYP
02590         INLCSA(INSECT,7)=ILENCD*PMULT
02600         INLCSA(INSECT,8)=IRES
02610         INLCSA(INSECT,9)=1000*IPHSE+ITT
02620         INLCSA(INSECT,10)=ISECD
02630         INLCSA(INSECT,11)=IZNGR
02640         INLCSA(INSECT,13)=IPRIM*100
02650         INLCSA(INSECT,14)=NLINE
02660         IF(IPAR.EQ.0)GOTO220
02670         INSEQV(IPAR)=INSEQV(IPAR)+1
02680         IF(INSEQV(IPAR).LE.2)GOTO240
02690         WRITE(06,10009)NLINE,IPAR
02700         IABORT=1
02710         GOTO190
02720    220  IF(JROOT.EQ.0)GOTO230
02730         WRITE(06,10010)NLINE,INSECT,JROOT
02740         IABORT=1
02750         GOTO190
02760    230  JROOT=INSECT
02770    240  IF(ILSON.EQ.0)GOTO250
02780         ISTAK(ILSON)=ISTAK(ILSON)+1
02790         IF(ISTAK(ILSON).EQ. 1)GOTO250
02800         WRITE(06,10011)NLINE,ILSON
02810         IABORT=1
02820         GOTO190
02830    250  IF(IRSON.EQ.0)GOTO190
02840         ISTAK(IRSON)=ISTAK(IRSON)+1
02850         IF(ISTAK(IRSON).EQ.1)GOTO190
02860         WRITE(06,10011)NLINE,IRSON
02870         IABORT=1
02880         GOTO190
02890    260  CONTINUE
02900 *         THIS SEGMENT IS FOR CONSISTENCY CHECKING
02910 *         IT GOES IN AFTER LINE1470 ALL MATRICES DEFINED IN MAIN
02920         DO 270 I=1,MAXND
02930         ISTAK(I)=-10
02940    270  CONTINUE
02950         DO 360 I=1,MAXND
02960         ND=I
02970         IF(INLCSA(I,1).LT.0)GOTO350
02980         ILSON=INLCSA(I,3)
02990         IRSON=INLCSA(I,4)
03000         IF(ILSON.NE.0.AND.INLCSA(ILSON,1).LT.0)GOTO320
03010         IF(IRSON.NE.0.AND.INLCSA(IRSON,1).LT.0)GOTO330
03020         IF(INLCSA(I,3).NE.0.OR.INLCSA(I,4).NE.0)GOTO360
03030         K=1
03040         ND=I
03050    280  INSEQV(K)=ND
03060         NPAR=INLCSA(ND,2)
03070         IF(NPAR.EQ.0)GOTO350
03080         IF(INLCSA(NPAR,3).EQ.0.AND.INLCSA(NPAR,4).EQ.0)GOTO290
03090         IF(INLCSA(NPAR,3).NE.ND.AND.INLCSA(NPAR,4).NE.ND)GOTO300
```

```
03100          ISTAK(ND)=10
03110          K=K+1
03120          ND=NPAR
03130          IF(K.GT.NOMAX)GOTO310
03140          GOTO280
03150     290  IF(ISTAK(ND).EQ.5)GOTO360
03160          ISTAK(ND)=5
03170          IABORT=1
03180          WRITE(6,10013)ND,INLCSA(ND,14),NPAR,INLCSA(NPAR,14)
03190          GOTO360
03200     300  IF(ISTAK(ND).EQ.5)GOTO360
03210          ISTAK(ND)=5
03220          IABORT=1
03230          WRITE(42,10014)ND,INLCSA(ND,14),NPAR,INLCSA(NPAR,14)
03240          GOTO360
03250     310  ISTAK(I)=5
03260          IABORT=1
03270          WRITE(42,10015)I,(INSEQV(J),J=1,K)
03280          GOTO360
03290     320  IESON=ILSON
03300          GOTO340
03310     330  IESON=IRSON
03320     340  ISTAK(I)=5
03330          IABORT=1
03340          WRITE(42,10012)INLCSA(I,14),I,IESON
03350          GOTO360
03360     350  ISTAK(ND)=10
03370     360  CONTINUE
03380          DO 370 I=1,MAXND
03390          IF(I.EQ.1)K=0
03400          IF(ISTAK(I).GE.0)GOTO370
03410          K=K+1
03420          INSEQV(K)=I
03430     370  CONTINUE
03440          IF(K.EQ.0)GOTO380
03450          WRITE(42,10016)(INSEQV(J),INLCSA(INSEQV(J),14),J=1,K)
03460     380  IF(IABORT.NE.1)GOTO400
03470          DO 390 I=1,MAXND
03480          IF(I.EQ.1)K=0
03490          IF(ISTAK(I).EQ.10)GOTO390
03500          K=K+1
03510          INSEQV(K)=I
03520     390  CONTINUE
03530          WRITE(42,10017)(INSEQV(J),INLCSA(INSEQV(J),14),J=1,K)
03540          STOP 197
03550     400  DO 410 I=1,MAXND
03560          INSEQV(I)=0
03570     410  ISTAK(I)=0
03580          MCOL=14
03590          CALL PROSEQ(INLCSA,INSEQV,ISTAK,MAXND,NTEST,MCOL,JROOT)
03600 *        PRINT,'AT128+3 NTEST=',NTEST
03610          XLENTH=0.0
```

```
03620 *       SECTION LEVELS AND PROCESSING SEQUENCE VECTOR HAVE BEEN CREATED
03630 *       THE NETWORK HAS BEEN EXHAUSTIVELY TESTED FOR CONSISTENCY
03640 *       THE NEXT STEP IS TO CREATE A DIRECTORY FOR WRITING THIS
03650 *       INFORMATION INTO AN OUTPUT FILE
03660 *       NUMBER OF 320 WORD BLOCKS IS 32+1.6 TIMES NUMBER OF SECTIONS.
03670   420 ENCODE(MDPRL,10018)IFCODE+1000,ITEMP,IASN,NIDEN
03680       NOMAX=NTEST
03690       ICBUF(1)=2
03700       ICBUF(2)=200+(NTEST+18)*MREC/320
03710       ICBUF(3)=ICBUF(2)
03720       ICBUF(4)=1
03730       ICBUF(5)=3
03740       IRECD=1
03750       CALL MOPENF(LDPRL,MDPRL,ISTAT,3,1,ICBUF,5,0)
03760       PRINT,'IF YOU WANT AN ASCII OUTPUT FILE ENTER 1,ELSE CR'
03770       READ,NOUTFL
03780       IF(NOUTFL.EQ.0)GOTO430
03790       KOUT=KOUT+1
03800       ENCODE(MPRS,10019)KOUT
03810       CALL OPENF(LOUT,MPRS,ISTAT)
03820       IF(ISTAT.NE.0)CALL FILSTO(MPRS,'STATUS',0,ISTAT,158)
03830   430 CALL RANSIZ(LDPRL,MREC,1)
03840       IF(NOUTFL.NE.0)PRINT,'OUTFILE NAME= ',MPRS
03850 ,     PRINT,'NETWORK FILE= ',MDPRL
03860       GOTO450
03870   440 WRITE(06,10024)LDPRL,IRECD
03880 *       FOR EXPLANATION OF ABOVE SYMBOLS SEE NOTES ON DATA FILE NTWIJKLM
03890 *       THE FOLLOWING SECTION MAKES UP THE DIRECTORY COLUMN IN
03900 *       INLCSA ARRAY
03910 *       THE FIRST SECTION WILL START AT RECORD 21 AND THEY WILL BE STORED
03920 *       IN THE SEQUENCE INDICATED BY INSEQV.
03930   450 DO 470 I=1,NOMAX
03940       NOSEQ=INSEQV(I)
03950       IF(INLCSA(NOSEQ,1).EQ.NOSEQ)GOTO460
03960       WRITE(06,10025)NOSEQ
03970       STOP 1021
03980   460 INLCSA(NOSEQ,14)=I+20
03990   470 CONTINUE
04000 *       INLCSA AND INSEQV ARE NOW WRITTEN ONTO DISK.  INLCSA OCCUPIES
04010 *         PRINTING CONTENTS FOR FIRST(ARBITRARILY)25 SECTIONS
04020 *       DO 475 I=1,25
04030 *       WRITE(06,10026)(INLCSA(I,J),J=1,14),INSEQV(I)
04040 * 475 CONTINUE
04050 *       IF(NOUTFL.NE.0)WRITE(LOUT,10027IFCODE+1000,ITEMP,IASN,NIDEN
04060 *         RECORDS 2 THRU 15,INSEQV OCCUPIES RECORD NUMBER 18
04070       DO 480 J=1,14
04080       IRECD=J+1
04090       WRITE(LDPRL' IRECD,ERR=440)(INLCSA(NINDX,J),NINDX=1,MAXND)
04100 *       IF(NOUTFL.NE.0)WRITE(LOUT,10020)'INLCSA COLUMN',J,(INLCSA(IP,J),
04110 *      &                  IP=1,40)
04120   480 CONTINUE
04130       IRECD=18
```

```
04140        IF (NOMAX.EQ.MAXND)GOTO500
04150        DO 490 I=NOMAX+1,MAXND
04160    490 INSEQV(I)=0
04170    500 WRITE(LDPRL'IRECD,ERR=440)(INSEQV(J),J=1,MAXND)
04180  *     IF(NOUTFL.NE.0)WRITE(LOUT,10020)'INSEQV',(INSEQV(IP),IP=1,NOMAX)
04190  *       NOTE STATUS WORD IS MEANINGLESS UPTO THIS POINT.
04200  *       WE NOW REMOVE ONTWK FILE FROM AFT.
04210        CALL  DETACH(LDAIN,ISTAT,)
04220        IF(ISTAT.EQ.0)GOTO520
04230        WRITE(06,10029)
04240        READ,ITEST
04250        IF(ITEST.NE.0)GOTO510
04260        WRITE(06,10030)
04270    510 WRITE(06,10031)
04280  *       THE REST OF THIS PROGRAM WILL CONTAIN SEGMENTS WHICH WILL BE
04290  *       DEVELOPED LATER IN APPROPRIATE PLACES WE SHALL ASSUME
04300  *       THE AVAILABILITY OF THE RESTULS OF THESE SEGMENTS. APPROPRIATE
04310  *       NOTATION WILL BE MADE.
04320  *       THE LOOP BELOW WILL DETERMINE DPA QUANTITIES FOR THE
04330  *       NETWORK. WE FIRST OPEN THE NECESSARY RANDOM BINARY FILES
04340  *       DASECD,DAPRIM,DATRAN,ABCD
04350  *       THEIR LGU NUMBERS,SYMBOLS AND NAME CHARACTERS ARE:
04360  *         FIILE NAME      LOC. NUMBER      LGU SYMBOL      CHARAC.NAME.
04370  *         DASECD              11              LDSEC           MDSEC
04380  *         DPRIM               12              LPRIM           MPRIM
04390  *         DRBTR               14              LDRBTR          MDRBTR
04400  *         DPU                 25              LDPU            MDPU
04410  *         DTRANN              15              LTRAN           MTRAN
04420    520 CONTINUE
04430  *        IOMCD THE VARIABLE KEEPING TRACK OF DPU FILES
04440  *        IS NEXT INITIALIZED
04450        IOMCD=-1
04460  *       WE NOW BEGIN A MAJOR LOOP FOR ANALYZING THE NETWORK
04470  *       THE PROCESSING LOOP BELOW FOLLOWS THE LOGIC OF 'PPROCESSING
04480  *       ALGORITHM' FLOWCHART OF 8/15/77
04490  *       WE FIRST INITIALIZE THE INFORMATION STORAGE AREAS
04500        DO 530 I=1,MAXND
04510    530 LEVPT(I)=0
04520  *       WE SET ALL OUTPUT QUANTITIES TO ZERO SO THAT THOSE WHICH ARE NOT
04530  *       MODIFIED WILL ALWAYS PRINT OUT AS ZERO.
04540        DO 550 I=1,NDIM
04550        AM(I)=ZERO
04560        BM(I)=ZERO
04570        CM(I)=ZERO
04580        DM(I)=ZERO
04590        DO 550 J=1,NDIM
04600        APHI(I,J)=ZERO
04610        BPHI(I,J)=ZERO
04620        CPHI(I,J)=ZERO
04630        DPHI(I,J)=ZERO
04640        VTOPH(I,J)=ZERO
04650        YIN(I,J)=ZERO
```

```
04660         Y2K(I,J)=ZERO
04670         DO 540 K=1,5
04680     540 YSTORE(I,J,K)=ZERO
04690         YPRIM(I,J)=ZERO
04700     550 CONTINUE
04710         NSTAK=1
04720     560 PRINT 10033,IOPC,XDB,XANG
04730         PRINT 10053
04740         READ,I
04750         IF(I.EQ.0)GOTO570
04760         PRINT 10033
04770         READ,IOPC,XDB,XANG
04780         GOTO560
04790     570 CONTINUE
04800         IF(IOPC.NE.3)GOTO580
04810         ENCODE(TRANF,10068)TTRAN
04820         PRINT 10052,TTRAN
04830         PRINT 10053
04840         READ,I
04850         IF(I.EQ.0)GOTO580
04860         PRINT 10052
04870         READ,TTRAN
04880         GOTO570
04890     580 CONTINUE
04900 *       IF(IOPC.EQ.0.AND.NOUTFL.NE.0)WRITE(LOUT,10021)'TRANSFORMER SHORT',
04910 *      &                                              ' SECONDARY',IOPC
04920 *       IF(IOPC.EQ.1.AND.NOUTFL.NE.0)WRITE(LOUT,10021)'TRANSFORMER OPEN',
04930 *      &                                              ' SECONDARY',IOPC
04940 *       IF(IOPC.EQ.2.AND.NOUTFL.NE.0)WRITE(LOUT,10021)'TRANSFORMER ANAL',
04950 *      &                                              'YTICAL MODEL',IOPC
04960 *       IF(IOPC.GT.2.AND.NOUTFL.NE.0)WRITE(LOUT,10021)'TRANSFORMER SPECIAL',
04970 *      &                                              ' FILE/ARRAY',IOPC
04980 *       IF(IOPC.LT.0.AND.NOUTFL.NE.0)WRITE(LOUT,10021)'TRANSFORMER LOADING',
04990 *      &                                              ' IGNORED',IOPC
05000         IF(IOPC.LT.0)GOTO590
05010         PRINT 10065,XDB,XANG
05020         X=10.**(0.05*XDB)
05030         BETA=CMPLX(X*COS(XANG/RAD),X*SIN(XANG/RAD))
05040         PRINT 10066,BETA
05050 *       PRINT,'NSECT ITT LTYP'
05060         YNEU=ZERO
05070 *         BEGIN MAJOR LOOP
05080     590 DO 1180 IND=1,NOMAX
05090         NSECT=INSEQV(NOMAX+1-IND)
05100         LEVND=INLCSA(NSECT,5)
05110         NPAR=INLCSA(NSECT,2)
05120         ILSON=INLCSA(NSECT,3)
05130         IRSON=INLCSA(NSECT,4)
05140         LEVPAR=INLCSA(NPAR,5)
05150         LTYP=INLCSA(NSECT,6)
05160         LENGTH=INLCSA(NSECT,7)
05170         INDRHO=INLCSA(NSECT,8)
```

```
05180        ITTY=INLCSA(NSECT,9)
05190        ISECD=INLCSA(NSECT,10)
05200        IZNGR=INLCSA(NSECT,11)
05210        IF(IZNGR.GT.0)YNEU=AONE/IZNGR
05220 *      IF(NOUTFL.NE.0)WRITE(LOUT,10034)NSECT,ITTY,LTYP,,1*LENGTH
05230 *      PRINT 10035,NSECT,ITTY,LTYP
05240        IF(IASN.LT.3.AND.IZNGR.EQ.0)PRINT 10054,0250,NSECT,IASN,IZNGR
05250 *        NEXT WE READ DATA IN AND DETERMINE APPROPRIATE A,B,C,D,ETC.
05260 *        I.E. MODAL OR PHASE AND THEN PROCEED WITH ANALYSIS*
05270 *    ************************************************************************
05280 *        THIS SECTION RESERVED FOR LOGICAL SECTION FOR IMPLEMENTING    *
05290 *        OTHER OPTIONS FOR THE PRESENT ONLY OPTION 1                   *
05300 *        I.E. ALL QUANTITIES PHASE WILL BE IMPLEMENTED                 *
05310 *    ************************************************************************
05320 *        THE FOLLOWING SECTION IS FOR PHASE QUANTITIES ONLY OPTION
05330 *        ONLY FEEDER PROPAGATION WILL BE CONSIDERED AT FIRST
05340 *        HENCE WE EITHER HAVE A PRIMARY LOAD ON THE FEEDER,
05350 *        OR A SINGLE PHASE PRIMARY BRANCH.
05360 *        NOTE *  NO ABORT PROVISION IS BEING MADE HERE.
05370 *        THE FOLLOWING SECTION IS REPRESENDED IN THE BLOCK DIAGRAM BY THE
05380 *        BLOCK MARKED PROCESS NSECT,GENERATE YN(I,J)AND STORE INFO
05390 *        ON DISK.  OUTPUT WILL BE STORED ON LOCATION CORRESPONDING TO
05400 *        THE INSEQV VECTOR.  I.E. AT NSECT+20
05410        NREC=INLCSA(NSECT,14)
05420        IF(LTYP.LE.9)GOTO600
05430        IF(LTYP.LE.23)GOTO650
05440        IF(LTYP.LT.33)GOTO690
05450        IF(LTYP.LT.100)GOTO720
05460        IF(LTYP.LT.1000)GOTO760
05470        PRINT 10055,0260,NSECT,LTYP
05480        STOP 0590
05490 *        THIS SECTION IS FOR LTYP GE 1000-SUBNETWORKS WITH 3C PRIMARIES
05500 *        NECO=LCU,IE NUMBER OF EFFECTIVE CONDUCTORS UNCHANGED
05510 *      IF(NOUTFL.NE.0)WRITE(LOUT,10036)NSECT
05520 *        THE FOLLOWING SECTION IS FOR 1 PHASE TO 3 PHASE CONNECTIONS
05530 *          IE LTYP=1-9
05540 *          NECO=EFFECTIVE NUMBER OF CONDUCTORS ON SINGLE PHASE SIDE
05550 *          LCU=EFFECTIVE NUMBER OF CONDUCTORS ON THREE PHASE SIDE
05560    600 CONTINUE
05570 *          THIS LOGIC FOR 3 PHASE TO 1 PHASE.  NOTE THE LAST
05580 *          PRECEDING PROCESSED SECTION WILL BE AN ACTUAL FEEDER
05590 *          SO THAT NEUF, ETC WILL BE VALID FOR DETERMINING LOAD
05600 *          SIDE CONDITIONS
05610 *      IF(NOUTFL.NE.0)WRITE(LOUT,10037)NSECT
05620        IF(LTYP.GT.3.AND.NECO.NE.2)PRINT 10056,0275,NSECT,LTYP,NECO
05630        IF(IASN.LE.2.AND.NEUF.EQ.1.AND.NECO.NE.2)PRINT 10056,0274,NSECT,
05640       &                                                 LTYP,NECO,NEUF
05650        IF(IASN.GE.3.AND.NEUF.EQ.1.AND.NECO.NE.1)PRINT 10056,0276,NSECT,
05660       &                                                 LTYP,NECO,NEUF
05670        IF(IASN.GE.3.AND.NEUF.EQ.0.AND.NECO.NE.2)PRINT 10056,0277,NSECT,
05680       &                                                 LTYP,NECO,NEUF
05690        IF(IASN.LE.2.AND.NECO.NE.2)PRINT 10056,0279,NSECT,LTYP,NECO
```

```
05700        IF(NECO.GT.2)PRINT 10056,0280,NSECT,LTYP,NECO
05710        LCU=NECO+2
05720        IF(LTYP.GT.3)LCU=3
05730 *          LTYP.GT.3 IMPLIES DELTA CONNECTION, IE NO NEUTRAL
05740 *          THIS ASSUMPTION MAY HAVE TO REVISED IF NOT TRUE
05750 *      IF(NOUTFL.NE.0)WRITE(LOUT,10022)'AT 3938 LCU,NECO=',LCU,NECO
05760        DO 610 I=1,LCU
05770        DO 610 J=1,LCU
05780        VTOPH(I,J)=ZERO
05790        DPHI(I,J)=ZERO
05800        Y2K(I,J)=ZERO
05810        IF((I.GT.NECO).OR.(J.GT.NECO))GOTO610
05820        IF(YIN(I,J).EQ.ZERO)GOTO630
05830        Y2K(I,J)=YIN(I,J)
05840    610 CONTINUE
05850        IF(LTYP.GT.3)GOTO640
05860 *        HERE DPHI IS USED TEMPORARILY AS AN NECO*NECO SIZED
05870 *        TRANSPOSE OF VTOPH
05880 *        WITH ZERO FILLS. FIRST APPROPRIATE VTOPH IS CONSTRUCTED
05890        VTOPH(1,LTYP)=AONE
05900        DPHI(LTYP,1)=AONE
05910        IF(LCU.GT.3)VTOPH(2,LCU)=AONE
05920        IF(LCU.GT.3)DPHI(LCU,2)=AONE
05930    620 CALL CMTMPY(Y2K,VTOPH,YTEMP,LCU,LCU,LCU,NDIM)
05940        CALL CMTMPY(DPHI,YTEMP,YIN,LCU,LCU,LCU,NDIM)
05950 *          LCU=NUMBER OF EFFECTIVE CONDUCTORS ON SOURCE SIDE
05960 *          NECO=NUMBER OF EFFECTIVE CONDUCTORS ON LOAD SIDE
05970 *        1<=LTYP<=9
05980        WRITE(LDPRL'NREC)NSECT,LTYP,LCU,NECO,XLENTH,ZNULL,DPHI,VTOPH,
05990       &                    YIN,ZNULL,Y2K,ZNULL,VNULL,VNULL,VNULL,VNULL
06000 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'Y2K FOR LTYP.LE.9,LTYP,NPAR,NSECT='
06010 ***   &       LTYP,NPAR,NSECT
06020 ***    IF(NOUTFL.NE.0)CALL CMTPRT(Y2K,NECO,NDIM,LOUT)
06030 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'YIN FOR LTYP.LE.9,LTYP,NPAR,NSECT='
06040 ***   &       LTYP,NPAR,NSECT
06050 ***    IF(NOUTFL.NE.0)CALL CMTPRT(YIN,LCU,NDIM,LOUT)
06060 *          RESETTING NECO FOR PARENT CHECK AND LOGIC AT 530+1
06070        NECO=LCU
06080        GOTO1100
06090    630 WRITE(06,10038)NSECT
06100        PRINT 10056,0281,NSECT,LTYP,NECO
06110    640 MPH1=LTYP/2-1
06120 *          THIS SECTION IS FOR PHASE TO PHASE TAKE-OFFS
06130        MPH2=MOD((LTYP-MPH1+1),3)+1
06140 *          LTYP    MPH1    MPH2
06150 *           4       1       2      MPH1=PHASE ON 3 PHASE SIDE TO
06160 *           5       1       3      WHICH SINGLE PHASE HI IS CONNECTED
06170 *           6       2       3
06180 *           7       2       1      MPH2=PHASE ON 3 PHASE SIDE TO
06190 *           8       3       1      SINGLE PHASE LO IS CONNECTED
06200 *           9       3       2
06210        VTOPH(1,MPH1)=AONE
```

```
06220        VTOPH(2,MPH2)=AONE
06230        DPHI(MPH1,1)=AONE
06240        DPHI(MPH2,2)=AONE
06250        GOTO620
06260    650 CONTINUE
06270  *          THIS LOGIC FOR THREE PHASE TRANSPOSTION.  WILL USE
06280  *          CONTINUITY OF NEUF TO DETERMINE ACCEPTABLE LOAD
06290  *          SIDE CONDITIONS
06300  *        IF(NOUTFL.NE.0)WRITE(LOUT,10039)NSECT
06310  *          THIS SECTION IS FOR 3 PHASE TRANSPOSITIONS.
06320  *          LYTP CAN ONLY BE 12,13,OR 23.
06330  *             LTYP=12 PHASES 1&2 INTERCHANGED
06340  *             LTYP=13 PHASES 1&3 INTERCHANGED
06350  *             LTYP=23 PHASES 2&3 INTERCHANGED
06360        IF((LTYP.EQ.12).OR.(LTYP.EQ.13).OR.(LTYP.EQ.23))GOTO660
06370        WRITE(06,10040)NSECT
06380        PRINT 10056,0300,NSECT,LTYP
06390    660 CONTINUE
06400        IF((LTYP.EQ.13.OR.LTYP.EQ.23).AND.(NECO.LT.(NEUFE+3)))
06410       &    PRINT 10056,0303,NSECT,LTYP,NECO,NEUFE
06420        IF(LTYP.EQ.12.AND.(NECO.LT.(NEUFE+2)))PRINT 10056,0304,NSECT,
06430       &                                              LTYP,NECO,NEUFE
06440        LCU=NECO
06450        K1=LTYP/10
06460        K2=LTYP-K1*10
06470        DO 670 I=1,NECO
06480        DO 670 J=1,NECO
06490        Y2K(I,J)=YIN(I,J)
06500        DPHI(I,J)=ZERO
06510    670 VTOPH(I,J)=ZERO
06520        IF(LTYP.EQ.12)I=3
06530        IF(LTYP.EQ.13)I=2
06540        IF(LTYP.EQ.23)I=1
06550        DPHI(I,I)=AONE
06560        VTOPH(I,I)=AONE
06570        DPHI(K1,K2)=AONE
06580        DPHI(K2,K1)=AONE
06590        VTOPH(K1,K2)=AONE
06600        VTOPH(K2,K1)=AONE
06610    680 CONTINUE
06620        CALL CMTIRC(YIN,K1,K2,NDIM,NDIM)
06630  *       NUMBER OF EFFECTIVE COUNDUCTORS IN AND OUT IS EQUAL
06640  *       10<=LTYP<=23
06650        WRITE(LDPRL'NREC)NSECT,LTYP,LCU,NECO,XLENTH,ZNULL,
06660       &     DPHI,VTOPH,YIN,ZNULL,Y2K,ZNULL,VNULL,VNULL,VNULL,VNULL
06670  ***   IF(NOUTFL.NE.0)WRITE(LOUT,10022)'Y2K FOR LTYP.LE.23',LTYP,NPAR,NSECT=
06680  ***  &     ,LTYP,NPAR,NSECT
06690  ***   IF(NOUTFL.NE.0)CALL CMTPRT(Y2K,NECO,NDIM,LOUT)
06700  ***   IF(NOUTFL.NE.0)WRITE(LOUT,10022)'YIN FOR LTYP.LE.23',LTYP,NPAR,NSECT=
06710  ***  &          ,LTYP,NPAR,NSECT
06720  ***   IF(NOUTFL.NE.0)CALL CMTPRT(YIN,NECO,NDIM,LOUT)
06730        GOTO1100
```

```
06740    690 CONTINUE
06750 *        THIS SECTION IS FOR 3-2 PHASE AND 2-1 PHASE ZERO LENGTHS
06760 *          IE LTYP=24-32
06770 *         LINE TYPES 26-32 CHOSEN ARBITRARILY(BY R. WOODING)
06780 *          CHECK NECO (FROM PRECEDING LAST PROCESSED SECTION)
06790 *          BASED ON REQUIREMNTS OF THIS SECTION
06800 *          LTYP=24,25,26 INVOLVE 3 PHASE TO 2 PHASE
06810        IF((LTYP.GE.24.AND.LTYP.LE.26).AND.NECO.NE.(NEUFE+2))
06820       &     PRINT 10056,0310,NSECT,LTYP,NECO,NEUFE
06830        IF((LTYP.GE.27.AND.LTYP.LE.32).AND.NECO.NE.(NEUFE+1))
06840       &     PRINT 10056,0311,NSECT,LTYP,NECO,NEUFE
06850 *      PRINT,'AT 6666 NECO,NEUFE=',NECO,NEUFE
06860        DO 700 I=1, NDIM
06870        DO 700 J=1,NDIM
06880        Y2K(I,J)=YIN(I,J)
06890        DPHI(I,J)=ZERO
06900        VTOPH(I,J)=ZERO
06910    700 CONTINUE
06920        NDOWN=2
06930        IF(LTYP.LE.26)NDOWN=3
06940        IF(LTYP.EQ.24)KXFER(1)=1
06950        IF(LTYP.EQ.24)KXFER(2)=2
06960        IF(LTYP.EQ.25)KXFER(1)=1
06970        IF(LTYP.EQ.25)KXFER(2)=3
06980        IF(LTYP.EQ.26)KXFER(1)=2
06990        IF(LTYP.EQ.26)KXFER(2)=3
07000        IF(LTYP.EQ.27.OR.LTYP.EQ.29.OR.LTYP.EQ.32)KXFER(1)=2
07010        IF(LTYP.EQ.28.OR.LTYP.EQ.30.OR.LTYP.EQ.31)KXFER(1)=1
07020        DO 710 I=1, NDOWN-1
07030        VTOPH(I,KXFER(I))=AONE
07040    710 CONTINUE
07050        VTOPH(NDOWN,NDOWN+1)=AONE
07060        CALL CMTRAN(VTOPH,DPHI,NDIM,NDIM,NDIM)
07070        CALL CMTMPY(YIN,VTOPH,YTEMP,NDIM,NDIM,NDIM,NDIM)
07080        CALL CMTMPY(DPHI,YTEMP,YIN,NDIM,NDIM,NDIM,NDIM)
07090        XLENTH=0.0
07100        LCU=NECO+1
07110 *        24<=LTYP<=32
07120        WRITE(LDPRL'NREC)NSECT,LTYP,LCU,NECO,XLENTH,ZNULL,DPHI,VTOPH,
07130       &                  YIN,ZNULL,Y2K,ZNULL,VNULL,VNULL,VNULL,VNULL
07140 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'Y2K FOR LTYP.LE.33,LTYP,',
07150 ***    &                'NPAR,NSECT=',LTYP,NPAR,NSECT
07160 ***    IF(NOUTFL.NE.0)CALL CMTPRT(Y2K,NECO,NDIM,LOUT)
07170        NECO=LCU
07180 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'YIN FOR LTYP.LE.33,LTYP,',
07190 ***    &                'NPAR,NSECT=',LTYP,NPAR,NSECT
07200 ***    IF(NOUTFL.NE.0)CALL CMTPRT(YIN,NECO,NDIM,LOUT)
07210        GOTO1100
07220 *        THE FOLLOWING SECTION IS FOR RATIO BANK TRANSFORMERS
07230 *          ARBITRARILY 33.LE.LTYP.LE.99 WILL BE RESERVED FOR
07240 *          RATIO BANK TRANSFORMER TYPE DESIGNATION
07250    720 CONTINUE
```

```
07260 *       THE FOLLOWING LOGIC IS PREDICATED ON ALL RATIO TRANSFORMERS
07270 *       BEING ONE OR THREE IDENTICAL SINGLE PHASE (IN A BANK) TWO
07280 *       WINDING TRANSFORMERS
07290 *       IN AS MUCH AS THESE DEVICES OCCUR ONLY INFREQUENTLY ON A
07300 *       FEEDER. AN ASCII DATABASE FILE WILL BE USED
07310 *       AT THIS TIME LACKING MEASURED DATA, A TWO WINDING
07320 *       RLCN MODEL WILL BE USED.  THE ASCII DATABASE
07330 *       FILE WILL CONTAIN A HEADER RECORD NOT USED BY THIS PROGRAM
07340 *       WHICH CAN BE USED FOR ANNOTATION.  ALL OTHER RECORDS WILL
07350 *       EACH CONTAIN THE FOLLOWING DATA STRUCTURE IN THE ORDER SHOWN
07360 *         LINE NUMBER (FOR EDITING, NOT USED BY PROGRAM)
07370 *         IDENTIFICATION NUMBER
07380 *         WINDING RESISTANCE (BOTH WINDINGS REFERED TO PRIMARY) OHMS
07390 *         LEAKAGE INDUCTANCE (BOTH WINDINGS REFERRED TO PRIMARY) HENRIES
07400 *         SHUNT CAPACITANCE ACROSS PRIMARY TERMINALS-FARADS
07410 *         SHUNT CONDUCTANCE ACROSS PRIMARY TERMINALS-MHOS
07420 *         IDEAL TRANSFORMER TURNS RATIO (PRIMARY TO SECONDARY)
07430 *         ADDITIONAL ANNOTATION NOT USED BY PROGRAM
07440 *       THE SHUNT ELEMENTS WILL BE USED TO COMPUTE A SCALAR COMPLEX
07450 *       ADMITTANCE YRBTR AND SIMILARLY THE SERIES IMPEDANCE ZRBTR
07460 *       USING THE WINDING RESISTANCE AND INDUCTANCE.  NOTE MAGNETIZING
07470 *       INDUCTANCE IS NEGLECTED.  THE SHUNT CAPACITANCE IS A ZEROETH
07480 *       ORDER APPROXIMATION TO THE COMBINED EFFECTS OF BUSHINGS
07490 *       AND STRAY CAPACITY.
07500 *       THE NATURE OF THE RATIO TRANSFORMER INSTALLATION WILL BE
07510 *       INFERRED FROM THE NATURE OF THE DOWNSTREAM CONDUCTOR SITUATION
07520 *       AS IMPLIED BY NEUFO AND NECO, IE
07530 *             NEUFO NECO
07540 *               0     2      SINGLE PHASE DELTA
07550 *               0     3      THREE PHASE DELTA
07560 *               1     1      SINGLE PHASE AND NEUTRAL (IE SINGLE PHASE WYE)
07570 *               1     3      THREE PHASE WYE
07580 *       AT THIS TIME ONLY THE FIRST WILL BE IMPLEMENTED AND IT WILL
07590 *       BE ASSUMED THAT THE PRIMARY IS CONNECTED PHASE TO NEUTRAL, IE
07600 *       A SINGLE PHASE WYE
07610 *       THEREFORE DOWNSTREAM SHOULD HAVE NECO=2
07620 *       REMEMBER THAT IZNGR NEEDS TO BE SPECIFIED IN DNWKINIJ NETWORK
07630 *       FILE FOR IASN=1,3
07640       IF(NEUFO.EQ.0.AND.NECO.EQ.2)GOTO722
07650       PRINT,'RBTR LOGIC IMPLEMENTED ONLY FOR NEUFO=0,NECO=2',NEUFO,NECO
07660       STOP 0400
07670   722 CONTINUE
07680       IF(KDRBTR.NE.0)GOTO725
07690       CALL OPENF(LRBTR,MRBTR,ISTAT,1,0,1)
07700       IF(ISTAT.EQ.0)GOTO0724
07710       PRINT 10032,MRBTR,LRBTR,ISTAT
07720       STOP 192
07730   724 KDRBTR=1
07740   725 REWIND LRBTR
07750       READ(LRBTR,10028)
07760   726 READ(LRBTR,10028,END=727)I,ID,XR,XL,XC,XG,XN
07770       IF(ID-LTYP)726,728,726
```

```
07780    727 PRINT 10064,LTYP,MRBTR,NSECT
07790        STOP 727
07800    728 CONTINUE
07810        YRBTR=CMPLX(XG,XTPI*XFREQ*XC)
07820        ZRBTR=CMPLX(XR,XTPI*XFREQ*XL)
07830        Y11=YIN(1,1)
07840        Y22=YIN(2,2)
07850        Y12=(YIN(1,2)+YIN(2,1))/2.
07860        Y2K(1,1)=Y11
07870        Y2K(2,2)=Y22
07880        Y2K(1,2)=Y12
07890        Y2K(2,1)=Y12
07900        Y1=XN*XN*(Y11+Y22+Y12+Y12)/ZRBTR+Y11*Y22-Y12*Y12
07910        YIN(1,1)=YRBTR+(Y11*Y22-Y12*Y12)/(ZRBTR*Y1)
07920        Y1=XN/(Y1*ZRBTR)
07930        VTOPH(1,1)=(Y22+Y12)*Y1
07940        VTOPH(2,1)=-(Y11+Y12)*Y1
07950        LCU=1
07960        IF(IASN.EQ.2.OR.IASN.EQ.4)GOTO730
07970        VTOPH(1,2)=-VTOPH(1,1)
07980        VTOPH(2,2)=-VTOPH(2,1)
07990        CALL YADMNU(YIN,YNEU,LCU,NDIM)
08000        LCU=LCU+1
08010    730 CONTINUE
08020        NEUFO=1
08030 *        33<=LTYP<=99
08040        WRITE(LDPRL'NREC)NSECT,LTYP,LCU,NECO,VTOPH,YIN,Y2K,
08050       &                    ZNULL,ZNULL,YRBTR,ZRBTR,XN,
08060       &                    ZNULL,VNULL,VNULL,VNULL,VNULL
08070 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'RBTR VTOPH,Y2K,YIN,LTYP,NPAR,ZNULL
08080 ***   &                   NSECT=',LTYP,NPAR,NSECT
08090 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'USE ONLY FIRST COLUMN FOR VTOPH'
08100 ***    IF(NOUTFL.NE.0)CALL CMTPRT(VTOPH,NECO,NDIM,LOUT)
08110 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'RBTR,Y2K,NECO,LCU,NSECT=',NECO,LCU,
08120 ***    IF(NOUTFL.NE.0)CALL CMTPRT(Y2K,NECO,NDIM,LOUT)
08130 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'RBTR YIN,LTYP,NPAR,NSECT=',LTYP,NPA
08140 ***    IF(NOUTFL.NE.0)CALL CMTPRT(YIN,LCU,NDIM,LOUT)
08150        NECO=LCU
08160        GOTO1100
08170 *        THIS SECTION IS FOR 30 AND 10 LINES WITHOUT
08180 *        PARALLELING SECONDARIES
08190 *        100.LE.LTYP.LE.999
08200    760 CONTINUE
08210 *      IF(NOUTFL.NE.0)WRITE(LOUT,10044)LTYP,NSECT
08220 *        WE FIRST MAKE SURE THAT THE PROPER DPU FILE IS OPENED
08230 *        IOMCD IS THE VARIABLE WHICH KEEPS TRACK OF WHICH FILE IS
08240 *        OPEN.  IF THE PROPER DPU FILE IS NOT OPEN,WE CLOSE THE PRESENT
08250 *        DPU FILE AND OPEN THE PROPER ONE.  IOMCD IS INITIALLY -1
08260 *          NOTE THAT THE ONLY PARAMETER WHICH COULD CHANGE TO
08270 *          CAUSE A CHANGE IN FILE,HENCE OHMCODE IOMCD IS UNIQUE
08280        IF(IOMCD.EQ.INDRHO)GOTO800
08290        CALL DETACH(LDPU,ISTAT,)
```

```
08300        IOMCD=INDRHO
08310        ENCODE(MDPU,10045)IFCODE+1000,ITEMP,IOMCD,IASN
08320        CALL OPENF(LDPU,MDPU,ISTAT,1,1,1)
08330        IF(ISTAT.EQ.0)GOTO770
08340        WRITE(06,10032)MDPU,LDPU,ISTAT,MNTWK
08350    770 CALL RANSIZ(LDPU,NDPUSZ,1)
08360        READ(LDPU'1)JFC,JTM,JOM,JASN,NRCD,LREC,NTYP,JDIM,
08370       &             IREC,NDTE,XTIME,XFREQ
08380        IF(LREC.NE.NDPUSZ)CALL ERRSTT(CN,352,LREC,NDPUSZ,LDPU)
08390        IF(IFCODE.NE.JFC)CALL FILSTO(MDPU,'FCODE',IFCODE,JFC,IOMCD)
08400        IF(ITEMP.NE.JTM)CALL FILSTO(MDPU,'ITEMP',ITEMP,JTM,IOMCD)
08410        IF(IOMCD.NE.JOM)CALL FILSTO(MDPU,'OHMS',ITEMP,JTM,NSECT)
08420        IF(IASN.NE.JASN)CALL FILSTO(MDPU,'ASSUM',IASN,JASN,IOMCD)
08430 *          READ IN DIRECTORY VECTOR
08440        DO 780 J=2,NRCD
08450        K11=(J-2)*LREC+1
08460        K22=MIN0(NTYP-99,(J-1)*LREC)
08470        READ(LDPU'J)(IDIR(K00),K00=K11,K22)
08480        IF(K22.EQ.NTYP-99)GOTO790
08490    780 CONTINUE
08500    790 CONTINUE
08510 *      PRINT,'AT 5643 IREC,NTYP=',IREC,NTYP
08520 *      PRINT 888,(J,IDIR(J),J=1,NTYP-99)
08530 *        THE PROPER DPU FILE IS NOW OPENED AND CHECKED OUT
08540    800 CONTINUE
08550        IF(LTYP.GT.NTYP)CALL ERRSTT(CN,353,LTYP,NTYP,IREC)
08560        IR=IDIR(LTYP-99)
08570        IF(IR.LE.0)CALL ERRSTT(CN,354,LTYP,NTYP,IR)
08580 *          USING LTOP TO AVOID UNNECESSARY DPULIJKN READ WHEN NEW
08590 *          LTYP SAME AS OLD LTOP.  NOTE MUST NOT CHANGE ANY OF THE
08600 *          RETURNED VARIABLES WITH THIS STRATEGY
08610 *      IF(NOUTFL.NE.0)WRITE(LOUT,10022)'AT 5364 LTYP,LTOP=',LTYP,LTOP
08620        IF(LTYP.EQ.LTOP)GOTO810
08630        CALL DPUFRW(LDPU,IR,0,NDIM,LTOP,NRECO,NTOT,NUMF,NUMS,NEUS,NEUF,
08640       &        Z,Y,ZO,YO,S,SI,DL)
08650        IF(NECO.GT.4) PRINT 10056,0282,NSECT,LTYP,NRECO,NEUF
08660 *          ESTABLISHING NEUFE FOR USE WITH NECO CHECKING ELSEWHERE
08670        NEUFE=NEUF
08680        IF(IASN.GE.3)NEUFE=0
08690 *      IF(NOUTFL.NE.0)WRITE(LOUT,10022)'AT 5381 LTYP,LTOP,NRECO=',
08700 *     &                 LTYP,LTOP,NRECO
08710        IF(LTYP.NE.LTOP)CALL FILSTO(MDPU,'LINE TYPE',
08720       & LTYP,LTOP,IOMCD)
08730    810 CONTINUE
08740 *          DETERMINE IF AT TERMINUS WHERE NECO AND NEUFO NEEDS TO
08750 *          BE REDEFINED
08760        IF(ILSON.EQ.0.AND.IRSON.EQ.0)NECO=NRECO
08770        IF(ILSON.EQ.0.AND.IRSON.EQ.0)NEUFO=NEUF
08780        IF(NEUF.NE.NEUFO)PRINT 10057,0355,NSECT,LTYP,NEUF,NEUFO
08790        IF(LTOP.EQ.LTYP)GOTO850
08800 *          IF ABOVE TRUE NEED NOT REPEAT ZEROING
08810        IF(NRECO.EQ.NDIM)GOTO850
```

```
08820 *        ZERO ALL UNNECESSARY ELEMENTS
08830         IF(NRECO.LT.NDIM)GOTO820
08840         WRITE(06,10047)NRECO,NDIM,LTYP,MDPU
08850    820  CONTINUE
08860 *        BEGIN ZEROING
08870         DO 830 I=NRECO+1,NDIM
08880         DL(I)=ZERO
08890         DO 830 J=1,NDIM
08900         Z(I,J)=ZERO
08910         Y(I,J)=ZERO
08920         ZO(I,J)=ZERO
08930         YO(I,J)=ZERO
08940         S(I,J)=ZERO
08950         SI(I,J)=ZERO
08960    830  CONTINUE
08970         DO 840 I=1,NRECO
08980         DO 840 J=NRECO+1,NDIM
08990         Z(I,J)=ZERO
09000         Y(I,J)=ZERO
09010         ZO(I,J)=ZERO
09020         YO(I,J)=ZERO
09030         S(I,J)=ZERO
09040         SI(I,J)=ZERO
09050    840  CONTINUE
09060 *        WE ARE NOW READY TO PROCESS THIS INFORMATION.
09070 *             DETERMINE LOAD ON SEGMENT Y2K=YIN+YLOAD
09080 *        FIRST ZERO YLOAD
09090    850  DO 860 I=1,NDIM
09100         DO 860 J=1,NDIM
09110         YLOAD(I,J)=ZERO
09120    860  CONTINUE
09130 ***      IF(NOUTFL.NE.0)WRITE(LOUT,10022)'NORMAL MODE PROCESSING
09140 ***     &          LTYP,NPAR,NSECT=',LTYP,NPAR,NSECT
09150 ***      IF(NOUTFL.NE.0.AND.(ILSON.NE.0.OR.IPSON.NE.0))
09160 ***     &     WRITE(LOUT,10022)'DOWNSTREAM YIN,LTYP,NPAR,NSECT=',
09170 ***     &                      LTYP,NPAR,NSECT
09180 ***      IF(NOUTFL.NE.0.AND.(ILSON.NE.0.OR.IRSON.NE.0))
09190 ***     &     CALL CMTPRT(YIN,NECO,NDIM,LOUT)
09200 *          NOW RESET LCU=NRECO FOR CHECKING ON YIN,YRPIM,ETC
09210         LCU=NRECO
09220 *        IF(NOUTFL.NE.0)WRITE(LOUT,10022)'AT 6380 NRECO,NECO,LCU=',
09230 *    &                NRECO,NECO,LCU,NSECT
09240 *        DETERMINE YLOAD TOTAL.
09250 *          NOTE EVENTUALLY WANT BOTH YLOAD AND Y2K
09260 *          WHICH ACCOUNTS FOR SOME EXTRA LOGIC
09270 *        CHECK TO SEE IF THERE IS A PRIMARY LOAD
09280 *          REMEMBER LEAST SIGNIFICANT DIGIT RESEVERED FOR STATUS
09290         IPRIM=INLCSA(NSECT,13)/100
09300         ICON=IPRIM/1000
09310         IP=MOD(IPRIM,1000)
09320         IF(IPRIM.EQ.0)GOTO920
09330         IF(KPRIM.NE.0)GOTO870
```

```
09340       ENCODE(MPRIM,10048)IFCODE*1000,ITEMP
09350       CALL OPENF(LPRIM,MPRIM,ISTAT,1,1,1)
09360       IF(ISTAT.NE.0)CALL FILSTO(MPRIM,'STATUS',0,ISTAT,370)
09370       CALL RANSIZ(LPRIM,MPRISZ,1)
09380       READ(LPRIM'1)JFRE,JTP,JREC,IPRM,NDTF,XTIME
09390       IF(IFCODE.NE.JFRE)CALL FILSTO(MPRIM,'FREQ',IFCODE,JFRE,370)
09400       IF(ITEMP.NE.JTP)CALL FILSTO(MPRIM,'TEMPERATURE',ITEMP,JTP,370)
09410       KPRIM=1
09420    870 READ(LPRIM'IP+1)IPRI,NCU,IFLAG,I1,I2,I3,
09430      & ((YLOAD(I,J),I=1,NCU),J=1,NCU)
09440 *         IF FOLLOWING TRUE, THE HAVE NORMAL CASE, IE NO OPEN DELTA
09450 *     PRINT,'AT 6591 1111 NCU=',NCU,ICON,LCU,NUMF
09460       IF(ICON.EQ.0)GOTO900
09470 *         AT THIS POINT ONLY ICON=4,5,6,7,8,9 IS PERMITTED CORRESPONDING
09480 *         TO OPEN DELTA LOAD
09490       IF(ICON.LT.4.OR.ICON.GT.9)PRINT 10062,0355,NSECT,LTYP,
09500      &                                  NEUF,NUMF,IPRIM
09510 *         NOW CHECK AGAINST ANY ILLEGAL COMBINATION.  LOGIC COULD
09520 *         PROBABLY BE NEATER
09530 *         FOR OPEN DELTA MUST HAVE NEUF=1 AND NUMF=2,3
09540       IF(NEUF.EQ.0.OR.NUMF.EQ.1)PRINT 10062,0365,NSECT,LTYP,
09550      &                                  NEUF,NUMF,IPRIM
09560       IF(NUMF.EQ.3)GOTO880
09570 *         FOR ICON=5,6,8,9 MUST HAVE NUMF=3
09580       IF((ICON.EQ.5.OR.ICON.EQ.6.OR.ICON.EQ.8.OR.ICON.EQ.9)
09590      & .AND.NUMF.NE.3)PRINT 10062,0366,NSECT,LTYP,NEUF,NUMF,IPRIM
09600 *         BY INFERENCE
09610       IF((ICON.EQ.4.OR.ICON.EQ.7).AND.NUMF.LT.2)
09620      & PRINT 10062,0367,NSECT,LTYP,NEUF,NUMF,IPRIM
09630 *         FOR OPEN DELTA TRANSFORMERS NCU CAN BE 2 OR 3
09640    880 CONTINUE
09650       NCU=NUMF
09660       IF(ICON.EQ.4)GOTO900
09670       IF(ICON.LE.6)GOTO890
09680       Y1=Y(1,1)
09690       Y(1,1)=Y(2,2)
09700       Y(2,2)=Y1
09710    890 IF(ICON.EQ.7)GOTO900
09720 *         NOTE ABOVE THAT FOR ICON=4,7 COMPLETE FOR NUMF=2,3
09730 *         AT THIS POINT ICON CAN ONLY BE 5,6,8,9 AND WE KNOW NUMF=NCU=3
09740       CALL CMTIRC(YLOAD,2,3,NCU,NDIM)
09750       IF(ICON.EQ.6.OR.ICON.EQ.9)CALL CMTIRC(YLOAD,1,2,NCU,NDIM)
09760    900 CONTINUE
09770 *     PRINT,'AT 6619 IPRIM,ICON,IP=',IPRIM,ICON,IP
09780 *     CALL CMTPRT(YLOAD,NCU,NDIM,06)
09790 *         IF NEUTRAL NOT AT ZERO POTENTIAL, AUGMENT YLOAD
09800 *         WITH YN=1.0/IZNGR.  PREVIOUS TEST PREVENTS /0
09810       IF(IASN.EQ.3.OR.IASN.EQ.4)GOTO910
09820       IF(NCU.EQ.LCU)GOTO910
09830       IF(NEUF .EQ. 0)GOTO910
09840       CALL YADMNU(YLOAD,YNEU,NCU,NDIM)
09850       NCU=NCU+1
```

```
09860    910 CONTINUE
09870 *          NOW CHECK ON PRIMARY LOAD COMPATIBILITY (SEE 362+2)
09880        IF(NCU.NE.LCU)PRINT 10063,0369,NSECT,LTYP,LCU,NECO,NCU,IPRIM
09890        IF(IPRI.NE.IP)PRINT 10058,0368,NSECT,LTYP,IPRI,IP
09900 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'PRIMARY LOAD ',IP,NSECT='',IP,NSECT
09910 ***    IF(NOUTFL.NE.0)CALL CMTPRT(YLOAD,NCU,NDIM,LOUT)
09920 *          IOPC LOGIC TO FOLLOW IS TEMPORARY AND ALLOW CONTROLS
09930 *          WHETHER TRANSFORMER OPEN OR SHORT SECONDARY IS USED OR
09940 *          WHETHER TRANSFORMER LOADING IS IGNORED ALTOGETHER
09950    920 CONTINUE
09960        IF(IOPC.LT.0)GOTO1010
09970        IPHSE=ITTY/1000
09980        ITT=ITTY-1000*IPHSE
09990        IF(ITT.EQ.0)GOTO1010
10000        NCU=9999
10010        IF(IOPC .EQ. 3)GOTO940
10020        IF(ITRAN.NE.0)GOTO930
10030        ENCODE(MTRAN,10049)IFCODE
10040        CALL OPENF(LTRAN,MTRAN,ISTAT,1,1,1)
10050        IF(ISTAT.NE.0)CALL FILSTO(MTRAN,'STATUS',0,ISTAT,3710)
10060        CALL RANSIZ(LTRAN,NATRSZ,1)
10070        READ(LTRAN'1)JFRE,JREC,NDTE,XTIME
10080        IF(IFCODE.NE.JFRE)CALL FILSTO(MTRAN,'FREQ ',IFCODE,JFRE,3710)
10090        ITRAN=1
10100    930 CONTINUE
10110    940 CONTINUE
10120        IF(IPHSE.EQ.1.AND.NEUF.EQ.0)PRINT 10059,3711,NSECT,LTYP,ITTY,NEUF,
10130       &                                                        NUMF
10140        IF(IPHSE .LT. 0 .OR. IPHSE .GT.9)PRINT 10059,3710,NSECT,LTYP,ITTY,
10150       &                                                          NEUF,NUMF
10160        IF((IPHSE.EQ.2).AND.(NEUF.EQ.0.OR.NUMF.EQ.1))
10170       & PRINT 10059,3371,NSECT,LTYP,ITTY,NEUF,NUMF
10180        IF((IPHSE.EQ.3).AND.(NEUF.EQ.0.OR.NUMF.LT.3))
10190       & PRINT 10059,3372,NSECT,LTYP,ITTY,NEUF,NUMF
10200        IF((IPHSE.EQ.4.OR.IPHSE.EQ.7).AND.NUMF.EQ.1)
10210       & PRINT 10059,3373,NSECT,LTYP,ITTY,NEUF,NUMF
10220        IF((IPHSE.EQ.5.OR.IPHSE.EQ.6.OR.IPHSE.EQ.8.OR.IPHSE.EQ.9).
10230       & AND.NUMF.NE.3)PRINT 10059,3374,NSECT,LTYP,ITTY,NEUF,NUMF
10240        IF(IOPC.LE.2)GOTO950
10250 *          USE SPECIAL TABLE PROCEDURE
10260 *      PRINT,'AT 6943 1111 NCU,LCU,NECO=',NCU,LCU,NECO
10270        CALL TRANAD(YTPUT,IFCODE,ITT,LDTRA,TRANF,TRANAY,10,9)
10280 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'YTPUT FROM TRANAD,ITTY,NPAR,NSECT='
10290 ***   &      ,ITTY,NPAR,NSECT
10300 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10067)YTPUT
10310 *      PRINT,'AT 6945 * IFCODE,ITTY,YTPUT=',IFCODE,ITTY,YTPUT
10320        GOTO980
10330 *        THIS PROCEDURE USES ANALYTIC TRANSFORMER MODEL RECORD
10340 *          MUST CORRESPOND WITH DTRANN12 FILE INFORMATION
10350 *      IN THE FOLLOWING IOPC=0,1 ARE TREATED BY MODIFYING QLOAD
10360 *          NOTE THIS ENTAILS AVAILABILITY OF DTRANN12 FILE FOR THESE CASES
10370    950 IF(LODFLG.NE.0)GOTO970
```


C-2

```
10380    960 WRITE(06,10002)
10390        READ,PMAGMN,PMAGST,PANGMN,PANGST
10400        WRITE(06,10003)PMAGMN,PMAGST,PANGMN,PANGST
10410        READ,ITEST
10420        IF(ITEST.NE.0)GOTO960
10430        LODFLG=1
10440    970 ITREC=ITTY - IPHSE*1000+1
10450        IF(ITREC .LE. 0)PRINT 10060,3730,NSECT,LTYP,ITTY,ITREC
10460        PLMAG=UNIFM2(1,PMAGMN,PMAGST)/100.0
10470        PLANG=UNIFM2(1,PANGMN,PANGST)
10480        PLANG=AMOD(PLANG,360.)/RAD
10490        QLOAD=CMPLX(PLMAG*COS(PLANG),PLMAG*SIN(PLANG))
10500        IF(IOPC .EQ. 0)QLOAD=CMPLX(10000.,0)
10510        IF(IOPC .EQ. 1)QLOAD=ZERO
10520        CALL YTRANP(YTPUT,VSPRAT,QLOAD,ITT,LTRAN)
10530    980 YTPUT=YTPUT*BETA
10540        CALL YTRAMT(YTPUT,YTEMP,IPHSE,NDIM)
10550        NCU=NUMF
10560 *      PRINT,'AT 6964 1111 YTEMP='
10570 *      CALL CMTPRT(YTEMP,NCU,NDIM,06)
10580        IF(NEUF .EQ. 0)GOTO990
10590        IF(IASN.EQ.3.OR.IASN.EQ.4)GOTO990
10600        CALL YADMNU(YTEMP,YNEU,NCU,NDIM)
10610        NCU=NCU+1
10620    990 CONTINUE
10630 *      IF(NOUTFL.NE.0)WRITE(LOUT,10022)'AT 7080 NECO,NCU,LCU=',NECO,NCU,LCU
10640 *      PRINT,'AT 7081 * NECO,NCU,LCU=',NECO,NCU,LCU
10650        IF(NCU.NE.LCU)PRINT 10061,0374,NSECT,LTYP,LCU,NECO,NCU
10660 *      IF(NOUTFL.NE.0)WRITE(LOUT,10022)'AT 6075 NREC,IPRI=',NREC,IPRI
10670        IF(IFLAG.NE.0)PRINT 10058,0376,NSECT,LTYP,IPRI
10680        DO 1000 I=1,NECO
10690        DO 1000 J=1,NECO
10700   1000 YLOAD(I,J)=YLOAD(I,J)+YTEMP(I,J)
10710 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'TRANSFORMER LOAD,ITTY,NPAR,NSECT=',
10720 ***   &     ITTY,NPAR,NSECT
10730 ***    IF(NOUTFL.NE.0)CALL CMTPRT(YTEMP,NECO,NDIM,LOUT)
10740 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'YLOAD+YTRAN,ITTY,NPAR,NSECT=',ITTY,
10750 ***   &                                  NPAR,NSECT
10760 ***    IF(NOUTFL.NE.0)CALL CMTPRT(YLOAD,NECO,NDIM,LOUT)
10770 *      CALL CMDBAN(YTEMP,NECO,NDIM,200,06,WKAREA)
10780   1010 CONTINUE
10790 *          IF NOT AT TERMINUS,THEN NECO ASSOCIATED WITH YIN
10800 *          SHOULD BE EQUAL TO LCU
10810 *          IF AT TERMINUS DO NOT ADD YIN WHICH MAY HAVE GARBAGE
10820 *          ALSO NOTE IN THIS CASE NECO HAS BEEN DEFINED
10830        IF(ILSON.EQ.0.AND.IRSON.EQ.0)GOTO1030
10840        IF(NECO.NE.LCU)PRINT 10061,0375,NSECT,LTYP,LCU,NECO
10850        DO 1020 I=1,NECO
10860        DO 1020 J=1,NECO
10870   1020 Y2K(I,J)=YLOAD(I,J)+YIN(I,J)
10880        GOTO1050
10890   1030 DO 1040 I=1,NECO
```

```
10900         DO 1040 J=1,NECO
10910   1040  Y2K(I,J)=YLOAD(I,J)
10920   1050  CONTINUE
10930 ***     IF(NOUTFL.NE.0)WRITE(LOUT,10022)'Y2K,LTYP,NPAR,NSECT=',
10940 ***    &                LTYP,NPAR,NSECT
10950 ***     IF(NOUTFL.NE.0)CALL CMTPRT(Y2K,NECO,NDIM,LOUT)
10960 *         LOAD HAS BEEN CREATED AND CAN NOW BE USED IN PROCESSING
10970 *         THE NEXT STEP IS TO CREATE LINE PARAMETERS
10980         IF(LENGTH.LT.0)PRINT 10055,3375,NSECT,LTYP,LENGTH
10990         IF(LENGTH.GT.0)GOTO1080
11000         DO 1060 I=1,NECO
11010         BM(I)=ZERO
11020         CM(I)=ZERO
11030         DO 1060 J=1,NECO
11040         APHI(I,J)=ZERO
11050         BPHI(I,J)=ZERO
11060         CPHI(I,J)=ZERO
11070         DPHI(I,J)=ZERO
11080         VTOPH(I,J)=ZERO
11090         YIN(I,J)=Y2K(I,J)
11100   1060  CONTINUE
11110         DO 1070 I=1,NECO
11120         AM(I)=AONE
11130         DM(I)=AONE
11140         APHI(I,I)=AONE
11150         DPHI(I,I)=AONE
11160         VTOPH(I,I)=AONE
11170   1070  CONTINUE
11180         GOTO1090
11190   1080  X=LENGTH/10.0
11200         CALL ABCDEN(DL,ZO,YO,S,SI,X,LCU,NDIM,AM,BM,
11210        & APHI,BPHI,CPHI,DPHI,YTEMP)
11220         CALL CMAPBC(APHI,BPHI,Y2K,VTOPH,LCU,LCU,LCU,NDIM)
11230         CALL CMTINV(VTOPH,LCU,LCU,NDIM,WKAREA)
11240         CALL CMAPBC(CPHI,DPHI,Y2K,YTEMP,LCU,LCU,LCU,NDIM)
11250         CALL CMTMPY(YTEMP,VTOPH,YIN,LCU,LCU,LCU,NDIM)
11260 *         IN THIS CASE WE ALSO COMPUTE OTHER MATRICES
11270   1090  CONTINUE
11280 *       IF(NOUTFL.NE.0)WRITE(LOUT,10022)'AT 6405 NSECT,NREC=',NSECT,NREC
11290 *           NO CHANGE IN NUMBER OF EFFECTIVE CONDUCTORS IN AND OUT
11300 *         100<=LTYP<=999
11310         WRITE(LDPRL'NREC)NSECT,LTYP,LCU,NECO,XLENTH,ZNULL,AM,BM,CM,
11320        & APHI,BPHI,CPHI,DPHI,VTOPH,YLOAD,ZNULL,YIN,ZNULL,Y2K,
11330        & VNULL,VNULL,VNULL,VNULL,QLOAD,VSPRAT
11340 *       IF(NOUTFL.NE.0)WRITE(LOUT,10021)'Y2K',NSECT
11350 *       IF(NOUTFL.NE.0)CALL CMTPRT(Y2K,NECO,NDIM,LOUT)
11360 *         THIS SECTION FOLLOWS PROCESSISNG ALGORITHM(8/15/77)
11370 *         AFTER PROCESS NSECT
11380   1100  CONTINUE
11390 *       IF(NOUTFL.NE.0)WRITE(LOUT,10021)'VTOPH',NSECT
11400 *       IF(NOUTFL.NE.0)CALL CMTPRT(VTOPH,LCU,NDIM,LOUT)
11410 ***     IF(NOUTFL.NE.0)WRITE(LOUT,10022)'YIN,LTYP,NPAR,NSECT=',
```

```
11420 ***  &                LTYP,NPAR,NSECT
11430 ***    IF(NOUTFL.NE.0)CALL CMTPRT(YIN,LCU,NDIM,LOUT)
11440        IF(NPAR.GT.0 )GOTO1110
11450        IF(IND.EQ.NOMAX)GOTO1180
11460        WRITE(06,10050)NSECT,IND
11470  1110  IF(LEVPAR.EQ.LEVND)GOTO1180
11480 *          THE ABOVE BRANCH OCCURS WHEN THE NEXT SECTION TO
11490 *          BE PROCESSED IS THE PARENT TO THE CURRENT SECTION
11500        IF(LEVPAR.LT.LEVND)GOTO1120
11510        WRITE(06,10051)NPAR,LEVPAR,NSECT,LEVND
11520  1120  IF(LEVPT(LEVND).LT.1)GOTO1160
11530 *          THIS LOGIC PICKS UP SAVED SIBLING YIN AND
11540 *          CHECKS WITH EXPECTED NECO
11550        LEVPT(LEVND)=0
11560        NSTAK=NSTAK-1
11570 *      PRINT,'AT 8061 NSECT,NECO,NSTAK',NSECT,NECO,NSTAK
11580        IF(NECOV(NSTAK).EQ.NECO)GOTO1130
11590        PRINT,'NSECT,NECO,NSTAK,NECOV(NSTAK)='
11600        PRINT,NSECT,NECO,NSTAK,NECOV(NSTAK)
11610        PRINT,'CHECK NECO CONSISTANCY ON BROTHER OF NSECT=',NSECT
11620  1130  CONTINUE
11630        NPAR=INLCSA(NSECT,2)
11640 *      IF(NOUTFL.NE.0)WRITE(LOUT,10020)'NSECT,NPAR,LSON,RSON,NSTAK,',
11650 *     &                'YIN,YSTORE,YSUM',
11660 *     &       NSECT,NPAR,INLCSA(NPAR,3),INLCSA(NPAR,4),NSTAK
11670        DO 1140 I=1,NECO
11680        YDIAG1(I)=YIN(I,I)
11690        YDIAG2(I)=YSTORE(I,I,NSTAK)
11700        DO 1140 J=1,NECO
11710 ***    YTEMP(I,J)=YSTORE(I,J,NSTAK)
11720        YIN(I,J)=YIN(I,J)+YSTORE(I,J,NSTAK)
11730  1140  CONTINUE
11740 *      IF(NOUTFL.NE.0)CALL CVDBAN(YDIAG1,NECO,NDIM,200,LOUT,WKAREA)
11750 *      IF(NOUTFL.NE.0)CALL CVDBAN(YDIAG2,NECO,NDIM,200,LOUT,WKAREA)
11760        DO 1150 I=1,NECO
11770  1150  YDIAG1(I)=YIN(I,I)
11780 *      IF(NOUTFL.NE.0)CALL CVDBAN(YDIAG1,NECO,NDIM,200,LOUT,WKAREA)
11790 ***    NSIB=INLCSA(NPAR,3)
11800 ***    IF(NSIB.EQ.NSECT)NSIB=INLCSA(NPAR,4)
11810 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'SIBLING YIN,NPAR,NSIB,NSECT=',
11820 ***   &      NPAR,NSIB,NSECT
11830 ***    IF(NOUTFL.NE.0)CALL CMTPRT(YTEMP,NECO,NDIM,LOUT)
11840 ***    IF(NOUTFL.NE.0)WRITE(LOUT,10022)'TOTAL YIN,PARENT,SIBLINGS=',
11850 ***   &      NPAR,NSIB,NSECT
11860 ***    IF(NOUTFL.NE.0)CALL CMTPRT(YIN,NECO,NDIM,LOUT)
11870        GOTO1180
11880  1160  LEVPT(LEVND)=1
11890 *          THIS LOGIC SETS ASIDE YIN AND CURRENT NECO FOR LATER RECALL
11900        NECOV(NSTAK)=NECO
11910 *      PRINT,'AT 8141 NSECT,NECO,NSTAK',NSECT,NECO,NSTAK
11920        DO 1170 I=1,NECO
11930        DO 1170 J=1,NECO
```

```
11940         YSTORE(I,J,NSTAK)=YIN(I,J)
11950         YIN(I,J)=ZERO
11960   1170 CONTINUE
11970         NSTAK=NSTAK+1
11980         IF(NSTAK.GT.49)PRINT,'NSTAK.GT.49  STOP'
11990   1180 CONTINUE
12000 *         END OF MAJOR LOOP
12010         WRITE(LDPRL'1,ERR=440)NIDEN,NDATE,WTIME,IFCODE,ITEMP,
12020        & IASN,NOMAX,MREC,NLCSA,NSEQV,NRDAT,MAXND,NECO,JROOT,
12030        & PMULT,IOPC,XDB,XANG
12040         NREC=INLCSA(JROOT,14)
12050         READ(LDPRL'NREC)NSECT,LTYP,NECO,NECO,XLENTH,ZNULL,AM,BM,CM,
12060        & APHI,BPHI,CPHI,DPHI,VTOPH,YLOAD,ZNULL,YIN,ZNULL,Y2K,
12070        & VNULL,VNULL,VNULL,VNULL,QLOAD,VSPRAT
12080         LTEMP=LOUT
12090         LOUT=42
12100         WRITE(LOUT,10021)'INPUT ADMITTANCE'
12110         WRITE(LOUT,10021)'DB-ANGLE'
12120         CALL CMDBAN(YIN,NECO,NDIM,200,LOUT,WKAREA)
12130         WRITE(LOUT,10021)'RECTANGULAR'
12140         CALL CMTPRT(YIN,NECO,NDIM,LOUT)
12150         CALL CMTCOP(YIN,YTEMP,NECO,NDIM)
12160         CALL CMTINV(YTEMP,NECO,NECO,NDIM,WKAREA)
12170         WRITE(LOUT,10021)'YIN INVERSE'
12180         CALL CMTPRT(YTEMP,NECO,NDIM,LOUT)
12190         DO 1200 I=1,NECO
12200         TEMP=ZERO
12210         DO 1190 J=1,NECO
12220         TEMP=TEMP+YIN(I,J)
12230   1190 CONTINUE
12240   1200 ZT(I)=TEMP
12250         WRITE(LOUT,10021)'CURRENT DRIVE FOR UNIT VOLTAGE'
12260         CALL CVEPRT(ZT,NECO,NDIM,LOUT)
12270         DO 1210 I=1,NECO
12280   1210 ZT(I)=AONE/YTEMP(I,I)
12290         WRITE(LOUT,10021)'SINGLE PHASE CURRENT DRIVE'
12300         CALL CVEPRT(ZT,NECO,NDIM,LOUT)
12310         LOUT=LTEMP
12320         CALL DETACH(LDPRL,ISTAT,)
12330         CALL DETACH(LDAIN,ISTAT,)
12340         CALL DETACH(LDPU,ISTAT,)
12350         CALL DETACH(LDSEC,ISTAT,)
12360         KTRSEC=0
12370         CALL DETACH(LTRAN,ISTAT,)
12380         ITRAN=0
12390         IF(NOUTFL.NE.0)PRINT,'REMEMBER WROTE OUTFILE ',MPRS
12400         IF(NOUTFL.NE.0)CALL DETACH(LOUT,ISTAT,)
12410         CALL DETACH(LPRIM,ISTAT,)
12420         KPRIM=0
12430 *       CALL DETACH(LRMTR,ISTAT,)
12440         KDRBTP=0
12450         CALL DETACH(LSUBN,ISTAT,)
```

```
12460        LTOP=-1
12470        NECO=-1
12480        PRINT,'TYPE 1 TO CONTINUE OR CR TO STOP'
12490        READ,I
12500        IF(I.EQ.0)STOP 9000
12510        GOTO100
12520 *          THE FOLLOWING PROCEDURE IS USED TO KEEP LOGICAL CHECK ON
12530 *          THE NUMBER OF EFFECTIVE CONDUCTORS TO PREVENT/DETECT
12540 *          POSSIBLE ERRORS IN LOGIC OR FILE RECORD INDEXING.
12550 *          LCU WILL REPRESENT THE NUMBER OF EFFECTIVE CONDUCTORS
12560 *           FOR THE COMPUTATION OF THE NEW YIN FOR THE CURRENT SECTION.
12570 *          NECO WILL REPRESENT THE NUMBER OF EFFECTIVE CONDUCTORS
12580 *           ASSOCIATED WITH THE NET YIN BEING PRESENTED TO THE CURRENT
12590 *           SECTION BY 'DOWNSTREAM' EFFECTS.
12600 *          NCU IS USED TO CHECK CONSISTANCY OF PRIMARY LOAD AND
12610 *           EXPANDED DISTRIBTUION TRANSFORMER LOADS
12620 *          EXCEPT FOR SINGLE PHASE TO THREE PHASE TRANSITIONS AND
12630 *          RATIO BANK TRANSFORMER SITUATIONS,AND OTHER THAN
12640 *          POSSIBLY ZERO LENGTH SECTIONS,LCU=NECO.
12650 *          A VECTOR NECOV(15)WILL BE USED TO KEEP TRACK OF NECO
12660 *          ASSOCIATED WITH THE SAVED YIN IN THE STACK
12670 *    DAPRYLIJ  RANDOM BINARY FILE CONTAINING PRIMARY LOAD DATA
12680 *    DARBTRLI  RANDOM BINARY FILE CONTAINING RATIO BANK TRANSFORMER DATA
12690 *    DASECDLI  RANDOM BINARY FILE CONTAINING SECONDARY LOAD DATA
12700 *    DATRANLI  RANDOM BINARY FILE CONTANING DISTRIBUTION TRANSFORMER DATA
12710 *    DNWKINIJ  ASCII FILE CONTAINING NETWORK LOGICAL DESCRIPTON
12720 *    ITRAN     FLAG TO CONTROL OPENING DTRANNIJ
12730 *    IAREC     RANDOM RECORD ADDRESS ASSOCIATED WITH FILE DARBTRLI
12740 *    IASN      ASSUMPTION CODE
12750 *    ICOND     INTEGER LENGTH CODE FROM DNWKINLI
12760 *    IDRBTR    RATIO BANK TRANSFORMER IDENT READ FROM DARBTRLI
12770 *    IFCODE    FREQUENCY CODE
12780 *    INDRHO    OHM CODE
12790 *    IOMCD     OHMCODE ASSOCIATED WITH CURRENT OPEN DPULIJKN FILE
12800 *    IOPC      TEMPORARY CONTROL FLAG FOR OPEN VS SHORT VS DISREGARD TRANS
12810 *    IPHS      PHASE CONNECTION CODE
12820 *    IPRIM     PRIMARY LOAD CODE
12830 *    IRECD     RANDOM RECORD ADDRESS ASSOCIATED WITH FILE NTLIJKNM
12840 *    IRES      OHM CODE FROM DNWKINLI
12850 *    IRTRAN    DISTRIBUTION TRANSFORMER TYPE CODE PLUS 1
12860 *    ISECD     SECONDARY LOAD TYPE CODE
12870 *    ISTM      ISTM+1=LAST RECORD IN DASECDLI FILE
12880 *    ITEMP     TEMPERATURE CODE
12890 *    ITTM      ITTM+1=LAST RECORD IN DATRANLI FILE
12900 *    ITTY      DISTRIBUTION TRANSFORMER TYPE CODE
12910 *    IZNGR     INTEGER GROUNDING ADMITTANCE
12920 *    JROOT     ROOT SECTION NUMBER
12930 *    KDRBTR    FLAG TO CONTROL OPENING DARBTRLI
12940 *    KPRIM     FLAG TO CONTROL OPENING DAPRYLIJ
12950 *    KTRSEC    FLAG TO CONTROL OPENING DASECDLI AND DATRANLI
12960 *    KVTSU     FLAG TO CONTROL OPENING SULIJKNM
12970 *    LCU       NUMBER OF EFFECTIVE CONDUCTORS ASSOCIATED WITH NEW YIN
```

```
12980 *    LDAIN=10  FRN FOR DNWKINIJ
12990 *    LDPRL=20  FRN FOR NTLIJKNM
13000 *    LDPU=25   FRN FOR DPULIJKN PER UNIT LENGTH DATABASE FILE
13010 *    LDSEC=11  FRN FOR DASECDLI SECONDARY LOAD DATABASE FILE
13020 *    LDTRA=13  FRN FOR TRANF USED IN TRANAD
13030 *    LENGTH    CODED LENGTH
13040 *    LEVND     LEVEL OF SECTION=INLCSA(NSECT,5)
13050 *    LEVPAR    LEVEL OF PARENT
13060 *    LOUT=16   FRN FOR WKEROUIJ
13070 *    LPRIM=12  FRN FOR DAPRYLIK
13080 *    LRBTR=14  FRN FOR DARBTRLI RATIO BANK TRANSFORMER DATABASE FILE
13090 *    LSUBN     FRN FOR ASCII SUBNETWORK INPUT PARAMTER FILE SUBNTLMN
13100 *    LTOP      LTYP ASSOCIATED WITH PREVIOUS READ OF DPULIJKN
13110 *    LTRAN=15  FRN FOR DATRANLI DISTRIBUTION TRANSFORMER DATABASE FILE
13120 *    LTYO      DUMMY TO BE DELTED FROM FINAL OPERATIONAL FORM
13130 *    LTYP      LINE TYPE CODE
13140 *    LVTSU     FRN FOR SULIJKNM
13150 *    MAXND=512 DIMENSION LIMIT TO MAXIMUM NUMBER OF SECTIONS
13160 *    MDPRL     FILENAME VARIABLE FOR NTLIJKNM
13170 *    MDPU      CHARACTER VARIABLE FOR FILENAME DPULIJKN
13180 *    MNTWK     FILENAME VARIABLE FOR DNWKINIJ
13190 *    MPRIM     CHARACTER VARIABLE FOR FILENAME DAPRYLIK
13200 *    MPRS      FILENAME VARIABLE FOR WKEROUIJ
13210 *    MRBTR     CHARACTER VARIABLE FOR FILENAME DARBTRLI
13220 *    MREC=760  RECORD SIZE FOR NTLIJKNM
13230 *    MTRAN     CHARACTER VARIABLE FOR FILENAME DATRANLI
13240 *    NCU       USED FOR PRIMARY LOAD DIMENSION CHECK
13250 *    NDIM=4    DIMENSION OF ADMITTANCE ARRAYS
13260 *    NDPUL     RECORD SIZE FOR DPULIJKN
13270 *    NDSEC     CHARACTER VARIABLE FOR FILENAME DASECDLI
13280 *    NECO      NUMBER OF EFFECTIVE CONDUCTORS ASSOCIATED WITH OLD YIN
13290 *    NEUF      NUMBER OF FEEDER NEUTRAL (0 OR 1), INDEPENDANT OF IASN
13300 *    NEUFE     NEUF MODIFIED BY IASN
13310 *    NEUFO     OLD VALUE OF NEUF, USED FOR CONTINUTIY CHECK
13320 *    NIDEN     NETWORK IDENTIFICATION NUMBER
13330 *    NLCSA=14  NUMBER OF RECORDS USED TO CONTAIN INLCSA ARRAY
13340 *    NOMAX     TOTAL NUMBER OF SECTIONS
13350 *    NPAR      PARENT SECTION=INLCSA(NSECT,2)
13360 *    NPRISZ=38 RECORD SIZE FOR DAPRYLIJ
13370 *    NRBTR=78  RECORD SIZE FOR DARBTRLI
13380 *    NRDAT=1   NUMBER OF RECORDS USED TO CONTAIN A SINGLE SECTION DATA
13390 *    NRECO     NUMBER OF EFFECTIVE CONDUCTORS READ IN FROM DPULIJKN
13400 *    NSECM     FILE DASECDLI CONTROL PARAMETER
13410 *    NSEQV=1   NUMBER OF RECORDS USED TO CONTAIN INSEQV VECTOR
13420 *    NTRNSZ    RECORD SIZE FOR DATRANLI
13430 *    NTRPH     PHASE CONNECTION CODE FOR DISTRIBUTION TRANSFORMER
13440 *    NVTSZ     RECORD SIZE FOR SULIJKNM
13450 *    SULIJKNM  RANDOM BINARY FILE CONTAINING COMPUTED SUBNET PARAMETERS
13460 *    TRANF     FILENAME FOR SPECIAL TRANSFORMER DATAFILE IOPC=3
13470 *    VTOPH     VOLTAGE TRANSFER RATIO MATRIX
13480 *    YIN       ADMITTANCE INTO THE SECTION
13490 *    Y2K       TOTAL SECTION TERMINATIN ADMITTANCE
```

```
13500 *    YLOAD     TOTAL EXTERNAL (IE NOT INCLUDING DOWNSTREAM YIN)LOAD
13510 *
13520 *
13530 *
13540 *         THE FOLLOWING ARE STOP CODE DIAGNOSTICS
13550 *
13560 *
13570 *      0250     IZNGR MUST NOT BE ZERO FOR IASN.LT.3
13580 *      0260     LTYP.GE.1000 NOT YET DEFINED
13590 *      0274     FOR IASN=1,2 AND NEUF=1, EXPECT NECO=2 FOR LTYP=1,2,3
13600 *      0275     EXPECT NECO=2 FOR LTYP=4,5,6,7,8,9, IE DELTA
13610 *      0276     FOR IASN=3,4 AND NEUF=1, EXPECT NECO=1 FOR LTYP =1,2,3
13620 *      0277     FOR IASN=3,4, AND NEUF=0 EXPECT NECO=2 FOR LTYP=4,5,6,7,8,9
13630 *      0279     FOR IASN=1,2 EXPECT NECO=2 FOR ANY LTYP=4,5,6,7,8,9
13640 *      0280     DO NOT EXPECT NECO.GT.2 FOR ANY LTYP=4,5,6,7,8,9
13650 *      0281     EXPECTED ALL NONZERO ELEMENTS IN Y2K FOR LTYP=4,5,6,7,8,9
13660 *      0282     NECO FROM THE DPU FILE IS GREATER THAN 4
13670 *      0300     EXPECTED LTYP=12,13,23
13680 *      0303     FOR LTYP=13,23 EXPECT NELO TO CORRESPOND TO THREE PHASE
13690 *      0304     FOR LTYP=12 NECO CANNOT CORRESPOND TO SINGLE PHASE
13700 *      0310     FOR LTYP=24,25,26 EXPECT NECO=NEUF+2
13710 *      0311     FOR LTYP=27,28,29,30,31,32 EXPECT NECO=NEUF+1
13720 *      0352     EXPECTED RECORD SIZE NOT EQUAL TO THAT IN DPUIJKLM
13730 *      0353     LTYP NOT CONTAINED IN DPUIJKLM
13740 *      0354     LTYP NOT DEFINED IN DPUIJKLM
13750 *      0355     EXPECT PRESENCE OR ABSENCE OF NEUTRAL TO REMAIN CONSTANT
13760 *      0355  1     EXPECT ACROSS RATIO BANKS (THIS TEST NOT NECESSARY
13770 *      0355  2     IF IASN=3,4
13780 *      0356     EXPECT ICON=4,5,6,7,8,9 FOR OPEN DELTA LOAD
13790 *      0365     FOR OPEN DELTA EXPECT NEUF=1 AND NUMF=2,3
13800 *      0366     FOR OPEN DELTA AND ICON=5,6,8,9 EXPECT NUMF=3
13810 *      0367     FOR OPEN DELTA AND ICON=4,7 EXPECT NUMF=2,3
13820 *      0368     PRIMARY LOAD SPECIFICATION IN DAPRYIJK INCONSISTANT
13830 *      0369     INCONSISTANCY BETWEEN LCU AND NCU WITH NCU DETERMINED
13840 *      0369  1     IN PRIMARY LOAD LOGIC AT 369-1, 3666, 365
13850 *      0374     INCONSISTANCY BETWEEN LCU AND NCU WITH NCU DETERMINED
13860 *      0374  1     IN DT-TTY LOGIC.  AT STATEMENT 3740+2 AND 373-1
13870 *      0375     FOR 100.LE.LTYP.LT.1000 LCU DETERMINED FROM DPUIJKLM
13880 *      0375  1     SHOULD EQUAL NECO DETERMINED FROM DOWNSTREAM
13890 *      0375  2     CONDITIONS
13900 *      0376     IFLAG FROM DAPRYIJK NOT ZERO; LOGIC/SUBROUTINE
13910 *      0376  1     NOT INSTALLED YET FOR PHYSICAL PARAMETERS
13920 *      3371     IN DT LOGIC WITH IPHSE=2 EXPECT NEUF=1 AND NUMF=2,3
13930 *      3372     IN DT LOGIC WITH IPHSE=3 EXPECT NEUF=1 AND NUMF=3
13940 *      3373     IN DT LOGIC WITH IPHSE=4,7 EXPECT NUMF=2,3
13950 *      3374     IN DT LOGIC WITH IPHSE=5,6,8,9 EXPECT NUMF=3
13960 *      3710     ILLEGAL IPHSE; EXPECT IPHSE=1,2,3,4,5,6,7,8,9
13970 *      3711     IPHSE EXPECTS DT TO BE CONNECTED PHASE TO NEUTRAL
13980 *      3730     ITTY IMPROPERLY DEFINED
13990        END
```

```
0010 *       THIS PROGRAM READS ASCII PRIMARY LOAD DATAFILES WITH FILENAMES
0020 *       OF TYPE AFPRYIJK
0030 *         IJ=IFREQ=2 DIGIT FREQUENCY CODE
0040 *          K=ITMP=1 DIGIT TEMPERATURE CODE
0050 *       THE STRUCTURE OF THIS FILE IS EXPLAINED BELOW
0060 *
0070 *       THIS PROGRAM CONSTRUCTS THE RANDOM BINARY FILE DAPRYIJK WITH
0080 *       A HEADER RECORD FOLLOWED BY DATA RECORDS
0090 *
0100 *       AFTER COMPLETING DAPRYIJK CONSTRUCTION, DAPRYIJK IS REREAD
0110 *       AND THE DATA USED TO REWRITE AFPRYIJK.  THE PURPOSE OF THIS
0120 *       IS TO CLEAN UP FORMAT AND PROVIDE A CHECK THAT DAPRYIJK HAS
0130 *       BEEN CONSTRUCTED PROPERLY.  THE ASCII FILE AFPRYIJK CAN BE
0140 *       PRINTED AND MODIFIED AND USED AS AN INPUT ON SUBSEQUENT
0150 *       REVISIONS
0160 *
0170 *       AN OPTION IS ALSO PROVIDED FOR EITHER LISTING, IOPT=1 OR
0180 *       CONVERTING DAPRYIJK TO AN ASCII FILE OUTFILNM, IOPT=2 WITHOUT
0190 *       CHANGING AFPRYIJK, IN ORDER TO CHECK CONTENTS
0200 *       OF DAPRYIJK.
0210 *
0220 *       NO ATTEMPT IS MADE TO PROVIDE SELECTIVE RECORD MAINTENANCE
0230 *       OR TO DETECT DUPLICATE ENTRIES
0240 *
0250 *       THE RANDOM BINARY FILE DAPRYIJK DATA RECORD HAS THE FOLLOWING
0260 *       SEQUENCE OF DATA
0270 *         IPRI=PRIMARY LOAD IDENTIFICATION NUMBER
0280 *         NECO=ANALYSIS DIMENSION
0290 *         IFLAG SPECIFIES ANALYTIC (IFLAG=0) OR PHYSICAL (IFLAG=1)
0300 *         YARY MATRIX ADMITTANCE NECO*NECO FOR IFLAG=0 OR
0310 *               NECO*(NECO+1)/2 PHYSICAL ELEMENTS
0320 *
0330 *       THE ASCII FILE AFPRYIJK HAS THE FOLLOWING MULTI RECORD
0340 *       FORMAT TO SPECIFY ONE SET OF DATA NOTE ILLUSTRATED FOR NECO=4
0350 *       FOR IFLAG=0:
0360 *
0370 *       LINE                              R=REAL PART
0380 *       LINE IPRI NECO  IFLAG             J=IMAGINARY PART
0390 *       LINE
0400 *       LINE YR11
0410 *       LINE YI11
0420 *       LINE
0430 *       LINE YR12  YR22
0440 *       LINE YI12  YI22
0450 *       LINE
0460 *       LINE YR13  YR23  YR33
0470 *       LINE YI13  YI23  YI33
0480 *       LINE
0490 *       LINE YR14  YR24  YR34  YR44
```

```
0500 *        LINE YI14  YI24  YI34  YI44
0510 *
0520 *        FOR IFLAG=1 A SIMILAR FORMAT IS USED WITH THE ENTRIES
0530 *        INTERPRETED AS PHYSICAL ELEMENTS
0540     10 FORMAT(/)
0550     11 FORMAT(A5,I2,I1,';')
0560     12 FORMAT('OUTFIL',I2,';')
0570     14 FORMAT(I4,2X,'IFRE=',I2,2X,'ITMP=',I1,2X,'LREC=',
0580    &       I2,2X,'IPRA=',I3,2X,'NDATE=',A9,2X,'TIME=',F6.2)
0590     15 FORMAT('IFRE=',I2,'  ITMP=',I1/'IF OK TYPE CR')
0600     16 FORMAT(I4,1P4E16.7)
0610     17 FORMAT(I4,3I5,8X,3I5)
0620        COMPLEX Y(4,4),ZEROA(4,4),Z,ZERO
0630        DIMENSION ICBUF(5),YR(4),YI(5)
0640        CHARACTER INFILE*9,OUFILE*9,SFP*6,NDATE*8
0650        DATA SFP/'SFPRYL'/,ZEROA/16*(0.,0.)/,ZERO/(0.,0.)/
0660        DATA ICBUF/2,9,9,1,3/,LUOU/12/,LREC/38/,KOUT/0/
0670        CALL MASTRK
0680        PRINT,'THIS PROGRAM READS AFPRYIJK AND CONSTRUCTS DAPRYIJK'
0690    100 CONTINUE
0700        PRINT,'TYPE CR FOR NORMAL FILE BUILDING OPTION IOPT=0'
0710        PRINT,'OR TYPE 1 BEFORE CR FOR IOPT=1 TO LIST DAPRYIJK'
0720        PRINT,'OR TYPE 2 BEFORE CR TO CONVERT DAPRYIJK TO'
0730        PRINT,'ASCII FILE OUTFILNM'
0740        READ,IOPT
0750        LUIN=0
0760        IF(IOPT.LE.1)GOTO105
0770        LUIN=11
0780        KOUT=KOUT+1
0790        ENCODE(INFILE,12)KOUT+100
0800        CALL OPENF(LUIN,INFILE)
0810 *        NOTE THAT NOW INFILE IS ASCII OUTFILNM
0820    105 CONTINUE
0830        PRINT,'TYPE TWO DIGIT FREQUENCY CODE, ONE DIGIT TEMPERATURE CODE'
0840        READ,IFRE,ITMP
0850        PRINT 15,IFRE,ITMP
0860        READ,I
0870        IF(I.NE.0)GOTO100
0880        ENCODE(OUFILE,11)'DAPRY',1000+IFRE,ITMP
0890        PRINT,OUFILE
0900    110 CONTINUE
0910 *        OPEN (CREATE IF NECESSARY) DAPRYIJK
0920        CALL OPENF(LUOU,OUFILE,ISTAT,3,1,ICBUF)
0930        IF(ISTAT.EQ.0)GOTO111
0940        PRINT,'UNABLE TO OPEN/CREATE ',OUFILE,' ISTAT=',ISTAT
0950        STOP
0960    111 CALL RANSIZ(LUOU,LREC,1)
0970        IF(IOPT.NE.0)GOTO402
0980        LUIN=11
0990        ENCODE(INFILE,11)'AFPRY',1000+IFRE,ITMP
1000        PRINT,INFILE
1010 *        OPEN BUT DO NOT CREATE AFPRYIJK
```

```
      CALL OPENF(LUIN,INFILE,ISTAT,3,0,1)
      IF(ISTAT.EQ.0)GOTO112
      PRINT,'ISTAT=',ISTAT
      IF(ISTAT.EQ.5)PRINT,INFILE,' DOES NOT EXIST'
      GOTO900
*       SKIP OVER SPACE OVER FIRST HEADER RECORD IN AFPRYIJK
  112 READ(LUIN,10)
*       NOW BEGIN READ AFPRYIJK, WRITE DAPRYIJK
      IPRM=0
  114 CONTINUE
*       READ NULL LINE FEED RECORD
      READ(LUIN,10,END=400)
*       READ NECO DATA RECORDS
      READ(LUIN,10,END=998)LINE,IPRI,NECO,IFLAG,I1,I2,I3
      DO 120 I=1,NECO
      READ(LUIN,10)
      READ(LUIN,10,END=998)LINE,(YR(J),J=1,I)
      READ(LUIN,10,END=998)LINE,(YI(J),J=1,I)
      DO 120 J=1,I
      Z=CMPLX(YR(J),YI(J))
      Y(I,J)=Z
      IF(I.EQ.J)GOTO120
      IF(IFLAG.EQ.0)GOTO115
*       IF IFLAG=1 SET UPPER   OUT TRIANGLE=0
      Y(J,I)=ZERO
      GOTO120
  115 Y(J,I)=Z
  120 CONTINUE
*       IF NECESSARY WRITE EMPTY RECORDS WITH IPRI=0
      IF(IPRI.EQ.IPRM)PRINT,'DUPLICATE IPRI=',IPRI
      IF(IPRI.LE.IPRM)GOTO240
  220 IF(IPRI.EQ.(IPRM+1))GOTO230
      IPRM=IPRM+1
      WRITE(LUOU'IPRM+1)0,0,0,ZERODA
      GOTO220
  230 IPRM=IPRI+1
  240 CONTINUE
*       NOW WRITE NORMAL RECORD.  NOTE THAT FOR IFLAG=0, UPPER-RIGHT
*       TRIANGLE IS ZERO
      WRITE(LUOU'IPRI+1)IPRI,NECO,IFLAG,I1,I2,I3,((Y(I,J),I=1,NECO),J=1,N
      GOTO114
  400 CONTINUE
*       AT THIS POINT END OF AFPRYIJK HAS BEEN REACHED NORMALLY
*       WRITE HEADER ON DAPRYIJK
      IF(IOPT.EQ.0)CALL DATIM(NDATE,TIME)
      WRITE(LUOU'1)IFRE,ITMP,LREC,IPRM,NDATE,TIME
*       NOW HAVE FINISHED CONSTRUCTION OF DAPRYIJK
*       NEXT REWRITE AFPRYIJK BY READING BACK DATA FROM DAPRYIJK
      REWIND(LUIN)
      ENDFILE(LUIN)
  402 READ(LUOU'1)IFRE,ITMP,LREC,IPRM,NDATE,TIME
      WRITE(LUIN,14)1000,IFRE+1000,ITMP,LREC,IPRM,NDATE,TIME
```

```
      LINE=1010
      NR=2
405   CONTINUE
      READ(LUIN,*,END=996)IPRI,NECO,IFLAG,I1,I2,I3,((Y(I,J),I=1,NECO),J=1,NECO)
      NR=NR+1
      IF(IPRI.LE.0)GOTO405
      WRITE(LUIN,17)LINE
      LINE=LINE+10
      WRITE(LUIN,17)LINE,IPRI,NECO,IFLAG,I1,I2,I3
      LINE=LINE+10
      DO 420 I=1,NECO
      DO 410 J=1,NECO
      Z=Y(I,J)
      YR(J)=REAL(Z)
410   YI(J)=AIMAG(Z)
      WRITE(LUIN,16)LINE
      LINE=LINE+10
      WRITE(LUIN,16)LINE,(YR(J),J=1,NECO)
      LINE=LINE+10
      WRITE(LUIN,16)LINE,(YI(J),J=1,NECO)
      LINE=LINE+10
420   CONTINUE
      IF(IPRI.LT.IPRO)GOTO405
      IF(IOPT.EQ.0)PRINT,OUFILE,' REWRITTEN'
      IF(IOPT.GT.1)PRINT,OUFILE,' CONVERTED TO ASCII AND PLACED',
     &          ' IN ',INFILE
500   PRINT,'TYPE 1 FOR NEW CASE'
      READ,I
      IF(I.EQ.0)STOP
      CALL DETACH(LUIN,ISTAT,)
      CALL DETACH(LUOU,ISTAT,)
      GOTO100
996   PRINT,'UNEXPECTED EOF ENCOUNTERED'
      STOP
      END
```

```
0010 *       THIS PROGRAM EXTRACTS FROM A SPECIFIED NETWORK FILE CONTAINED
0020 *       IN THE INFILE, A SUBNETWORK IE THAT PORTION INCLUDING AND
0030 *       BEYOND A SPECIFIED SECTION WHICH OCCURS SOMEWHERE WITHIN THE
0040 *       GIVEN NETWORK.  THIS PROGRAM ALLOWS GENERATION OF ANY NUMBER
0050 *       OF SUBNETWORKS.
0060 *       THIS PROGRAM INCORPORATES AN APPROACH SUGGESTED BY R WOODING
0070 *       NOTE THAT ASSIGNED SECTION NUMBERS MUST BE EACH LESS THAN THE
0080 *       DIMENSION ASSIGNMENT OF THE NETWORK ARRAY IA WHICH IS PRESENTLY
0090 *       IAL=2000
0100     10 FORMAT(V)
0110     11 FORMAT(I4,5I5,1X,7I5)
0120     12 FORMAT('PRINT NAME OF ',A6,' FOLLOWED BY A ;')
0130     13 FORMAT('IF OK TYPE CR, TYPE 1 TO CHANGE, TYPE -1 TO ESCAPE')
0140     14 FORMAT('NEW NETWORK ID NUMBER=',I2)
0150     15 FORMAT('NEW ROOT SECTION &=',I5)
0160        CHARACTER INFILE*9,OUFILE*9
0170        DIMENSION IA(2000,11),IS(200)
0180        DATA INFILE/'DNWKIN98;'/,OUFILE/'DNWKIN99;'/
0190        DATA IAL/2000/,ISL/200/,INET/99/,IROOT/100/
0200        DATA LUIN/11/,LUOU/12/
0210        CALL NASTRK
0220    130 CONTINUE
0230        PRINT,'INFILE=',INFILE
0240        PRINT 13
0250        READ,I
0260        IF(I)999,150,135
0270    135 PRINT 12,'INFILE'
0280        READ,INFILE
0290        GOTO130
0300    150 CONTINUE
0310        CALL OPENF(LUIN,INFILE,ISTAT,1,0,1)
0320        IF(ISTAT.EQ.0)GOTO160
0330        IF(ISTAT.EQ.5)PRINT,INFILE,' DOES NOT EXIST'
0340        IF(ISTAT.NE.5)PRINT,INFILE,' STATUS=',ISTAT
0350        GOTO130
0360 *         FIRST RESET FIRST COLUMN OF IAL TO -1 AS A FLAG
0370    160 DO 165 I=1,IAL
0380    165 IA(I,1)=-1
0390 *         NOW READ IN NETWORK ARRAY AND PLACE SECTION IN INSECT ROW
0400        READ(LUIN,10)IDUM,I
0410    170 READ(LUIN,10,END=180)IDUM,ISECT,(IA(ISECT,J),J=1,11)
0420        GOTO170
0430    180 CONTINUE
0440        PRINT,'OUFILE=',OUFILE
0450        PRINT 13
0460        READ,I
0470        IF(I)130,220,210
0480    210 PRINT 12,'OUFILE'
0490        READ,OUFILE
```

```
0500        GOTO180
0510    220 CONTINUE
0520        CALL OPENF(LUOU,OUFILE,ISTAT,3,0,1)
0530        IF(ISTAT.NE.5)GOTO230
0540        CALL OPENF(LUOU,OUFILE)
0550        GOTO250
0560    230 IF(ISTAT.NE.0)GOTO240
0570        PRINT,OUFILE,' ALREADY EXISTS, TYPE CR IF OK TO DESTROY/REWRITE'
0580        READ,I
0590        IF(I)180,250,180
0600    240 PRINT,OUFILE,' ISTAT=',ISTAT
0610        GOTO180
0620 *        AT THIS POINT OUTPUT FILE OPEN
0630    250 PRINT 14,INET
0640        PRINT 13
0650        READ,I
0660        IF(I)180,270,260
0670    260 PRINT 14
0680        READ,INET
0690        GOTO250
0700    270 CONTINUE
0710        PRINT 15,IROOT
0720        PRINT 13
0730        READ,I
0740        IF(I)250,290,280
0750    280 PRINT 15
0760        READ,IROOT
0770        GOTO270
0780    290 LINE=1000
0790        WRITE(LUOU,11)LINE,INET
0800        LINE=1001
0810        NSTACK=0
0820        ISECT=IROOT
0830 *        BEGINNING PROCESSING LOOP
0840 *        INDEX KEY
0850 *          1 IPAR
0860 *          2 ILSN
0870 *          3 IRSN
0880 *          4 LTYP
0890 *          5 ILCD
0900 *          6 IOHM
0910 *          7 ITTY
0920 *          8 IPHS
0930 *          9 NSER
0940 *         10 IPRL
0950 *         11 INGD
0960 *        REMEMBER SECTION DATA STORED BY ISECT ROW
0970 *        THE FOLLOWING IS A PROGRAMMING ERROR CHECK, REMOVE AFTER DEBUG
0980    295 IF(IA(ISECT,1).LT.0)STOP 0295
0990        IF(ISECT.NE.IROOT)GOTO320
1000 *        SAVE PARENT SECTION NUMBER FOR LATER RESTORATION
1010        IPAR=IA(IROOT,1)
```

```
      IA(IROOT,1)=0
  320 WRITE(LUOU,11)LINE,ISECT,(IA(ISECT,J),J=1,11)
      LINE=LINE+1
      ILSN=IA(ISECT,2)
      IRSN=IA(ISECT,3)
*      CHECK FIRST TO SEE IF A TERMINUS SECTION AND IF YES POP STACK
      IF(ILSN.EQ.0.AND.IRSN.EQ.0)GOTO500
*        NEXT CHECK TO SEE IF BRANCHING IS INVOLVED
      IF(ILSN.NE.0.AND.IRSN.NE.0)GOTO400
*        AT THIS POINT NO BRANCHING INVOLVED
      ISECT=MAX0(ILSN,IRSN)
      GOTO295
  400 CONTINUE
*        AT THIS POINT BRANCHING IS INVOLVED.  ARBITRARILY PLACE IRSN IN
*        STACK VECTOR IS
      IF(NSTACK.LT.ISL)GOTO420
      PRINT,'STACK EXHAUSTED. INCREASE IS DIMENSION'
      GOTO999
  420 NSTACK=NSTACK+1
      IS(NSTACK)=IRSN
      ISECT=ILSN
      GOTO295
  500 CONTINUE
*        AT THIS POINT A TERMINUS SECTION WAS REACHED.  LOOK AT
*        STACK VECTOR FOR A NEW SECTION NOT YET PROCESSED.  IF
*        NSTACK, THE NUMBER OF AS YET UNPROCESSED SECTIONS, IS
*        ZERO THEN WE ARE FINISHED, IF NOT ZERO, POP STACK
      IF(NSTACK.EQ.0)GOTO900
      ISECT=IS(NSTACK)
      NSTACK=NSTACK-1
      GOTO295
  900 CONTINUE
*        AT THIS POINT HAVE EXHAUSTED STACK AND FINISHED WITH
*        OUFILE.
      CALL DETACH(LUOU,ISTAT,)
      PRINT,'WROTE ',OUFILE,' WITH NETWORK NUMBER=',INET
      PRINT,'AND ROOT SECTION=',IROOT
*        NOW REPLACE PARENT NODE IN IA ARRAY
      IA(IROOT,1)=IPAR
      PRINT,'TYPE 1 TO GENERATE A NEW SUBNETWORK, TYPE CR TO ESCAPE'
      READ,I
      IF(I)130,130,180
  999 CALL DETACH(LUIN,ISTAT,)
      STOP 0999
      END
```

TRAPAD 4T 04/09/81 1:31 PM

```
010 *       THIS PROGRAM COMPUTES THE THREE PHASE ADMITTANCE ASSOCIATED
020 *       WITH A POWER FACTOR CORRECTION CAPACITOR BANK TRAPPED IN THE
030 *       COMMON RETURN BY AN INDUCTOR (SERIES R AND L) SHUNTED BY A
040 *       CONDUCTANCE GS.
050     10 FORMAT(V)
060     11 FORMAT('TYPE CR IF OK')
070     12 FORMAT('C=',1PE9.3)
080     13 FORMAT('R,GS,L=',1P3E12.3)
090     14 FORMAT('F=',F8.2)
100     15 FORMAT(72('*'))
110     16 FORMAT('TYPE CR TO CONTINUE')
120     17 FORMAT('Y11,Y12=',1P2E12.3,5X,2E12.3)
130        COMPLEX YC,ZT,Y11,Y12,ZONE,DEN
140        REAL L
150        DATA GS/.001/,R/9./,L/.004/,C/.2E-6/,F/5010./,ZONE/(1.,0.)/
160    100 PRINT 16
170        READ,I
180        IF(I.GE.0)STOP 0100
190    110 PRINT 13,R,GS,L
200        PRINT 11
210        READ,I
220        IF(I)100,120,115
230    115 PRINT 13
240        READ,R,GS,L
250        GO TO 110
260    120 PRINT 12,C
270        PRINT 11
280        READ,I
290        IF(I)110,130,125
300    125 PRINT 12
310        READ,C
320        GOTO120
330    130 PRINT 14
340        READ,F
350        IF(F.LE.0.)GOTO120
360        TPIF=6.2831851*F
370        ZT=CMPLX(R,TPIF*L)/(ZONE+GS*CMPLX(R,TPIF*L))
380        YC=CMPLX(0.,TPIF*C)
390        DEN=ZONE+3.*YC*ZT
400        Y11=YC*(ZONE+2.*YC*ZT)/DEN
410        Y12=-YC*YC*ZT/DEN
420        PRINT,'ZT=',ZT
430        PRINT,'YC=',YC
440        PRINT,'DEN=',DEN
450        PRINT 15
460        PRINT 14,F
470        PRINT 12,C
480        PRINT 13,R,GS,L
490        PRINT 17,Y11,Y12
```

```
500         PRINT 15
510         PRINT 16
520         READ,I
530         IF(I.NE.0)STOP 0130
540         GOTO130
550         END
```

```
**********************************************************************

UGZYGES1                                          04/10/81     3:05 PM

**********************************************************************


1000 *        THIS PROGRAM CONVERTS THE CONCENTRIC NEUTRALS OF U.G.
1010 *        POWER CABLES INTO ELECTRICALLY EQUIVALENT SINGLE NEUTRAL
1020 *        CONDUCTOR ANALOGOUS TO THE SITUATION FOUND IN OPEN WIRE
1030 *        OVERHEAD CONDUCTORS.THIS PROGRAM USES IMPEDANCE AND
1040 *        ADMITTANCE DATA IN UNIT OF OHMS/METER AND MHOS/METER
1050 *        CALCULATED FOR BURIED CABLE BY USE OF THE BPA EMTP
1060 *        CABLE CONSTANTS OPTION.
1070 *        IN CALCULATION PERFORMED BY THE PROGRAM THE ELECTRICAL
1080 *        EQUIVALENT NEUTRAL CONDUCTOR IS NOT ASSUMED TO BE AT
1090 *        GROUND POTENTIAL THUS RETAINING THE EVUIVALENT NEUTRAL
1100 *        AS AN 'EXPLICIT' CONDUCTOR.
1110 *          THIS PROGRAM IS EQUIPPED TO SUPPLY THE
1120 *        PROPER Z AND Y MATRICES FOR EITHER 1,2 OR 3 PHASE CASE BY
1130 *        THE PROPER EXTRACTION AND MANIPULATION OF THE ELEMENTS OF
1140 *        THE 6 BY 6 IMPEDANCE AND ADMITTANCE MATRICES SUPPLIED BY
1150 *        THE B.P.A. PROGRAM
1160 *        WRITTEN BY RICHARD WOODING
1170 *          4/7/81
1180 *          EXTENSIVE MODIFICATIONS WERE COMPLETED ON THE ORIGINAL
1190 *          NEUREDS1 PROGRAM DESCRIBED ABOVE.THIS PROGRAM NOW
1200 *          HAS THE CAPABILITY TO GENERATE Z & Y DATA FOR
1210 *          UNDERGROUND CABLES INTERNALLY AND THEN PERFORM THE
1220 *          NEUTRAL REDUCTION FUNCTION USING ALGABRAIC METHODS.
1230 *             THE PROCESSING OPTIONS NOW ARE;
1240 *
1250 *            1) USE OF PROPERLY FORMATTED E.M.T.P. ZY DATA.
1260 *
1270 *            2) MANUAL ENTRY OF Z & Y DATA SUPPLIED BY THE USER.
1280 *
1290 *            3) USE OF AN ASCII DATA FILE SUPPLYING 'LTYP' CASE
1300 *               INFORMATION AND PHYSICAL PARAMETERS FOR THE CABLE
1310 *               NECESSARY TO CALCULATE ZY DATA.
1320 *
1330 *
1340      10 FORMAT(V)
1350      20 FORMAT(I4,10I5,F11.3)
1360      30 FORMAT(I4,1X,1PE15.8,1X,E15.8)
1370      40 FORMAT(I4)
1380         NFN=9999
1390         DATA LUIN/11/,LUOU/12/,LUDF/13/
1400         CHARACTER*9 ROW,IMPED,ADMIT
1410         CHARACTER*9 INFILE,OUTFILE,CDFIL
1420         COMPLEX CC,CM1,CM2,CM3,CM4,ZMOD5,ZMOD6
1430 *         DATAZ IS THE IMPEDANCE MATRIX,DATAY IS THE ADMITTANCE MATRIX
1440         DIMENSION DATA(24,6),DATAZ(6,6),DATAY(6,6),ZMOD(4),ZMOD6(4),
1441        &          DATZ2(6,6),DATY2(6,6),DATZ3(6,6)
1450         COMPLEX DATAZ,DATAY,ZMOD,Q1,Q2,Q3,DATZ2,DATY2,DATZ3
1460         DIMENSION DENT(7)
1470         PRINT,'FOR USE OF AN EMTP FILE TYPE A CR'
```

```
      PRINT,'FOR USE OF MANUAL ENTRY TYPE A 1'
      PRINT,'FOR USE OF PHYSICAL DATA FILE TYPE A 2'
      READ,MPMODE
  100 CONTINUE
      IF(MPMODE.EQ.2)GOTO110
      IF(MPMODE.EQ.1)GOTO110
      PRINT,'ENTER THE NAME OF THE E.M.T.P. DATA FILE FOLLOWED BY;'
      READ,INFILE
      CALL OPENF(LUIN,INFILE,ISTAT,3,0,1)
      IF(ISTAT.NE.0)PRINT,'ISTAT=',ISTAT
  110 PRINT,'ENTER THE NAME OF THE OUTPUT FILE FOLLOWED BY A ;'
      READ,OUTFILE
      CALL OPENF(LUOU,OUTFILE,ISTAT,3,0,0)
      IF(MPMODE.EQ.2)GOTO140
      IF(MPMODE.EQ.1)GOTO120
      READ(LUIN,10,END=999)NPR1,NOCASE,NRC,NFQC,NCC,NTPC,NTOT,NPHSET,
     &                     NSP,NSN,NEUF
      READ(LUIN,10)LNUM0,IMPED
      PRINT,'LTYPE CASE DATA'
      PRINT,NPR1,NOCASE,NRC,NFQC,NCC,NTPC,NTOT,NPHSET,NSP,NSN,NEUF
      PRINT,'TYPE A CR IF OK'
      READ,NCHK
      IF(NCHK.EQ.0)GOTO130
  120 PRINT,'ENTER MODIFIED DATA CASE'
      READ,NPR1,NOCASE,NRC,NFQC,NCC,NTPC,NTOT,NPHSET,NSP,NSN,NEUF
      PRINT,'MODIFIED CASE DATA'
      PRINT,NPR1,NOCASE,NRC,NFQC,NCC,NTPC,NTOT,NPHSET,NSP,NSN,NEUF
      PRINT,'TYPE CR IF O.K.'
      READ,NCHK
      IF(NCHK.EQ.0)GOTO130
      IF(NCHK.NE.0)GOTO999
  130 CONTINUE
      IF(MPMODE.EQ.1)GOTO149
      LINX=1
      DO 135 I=1,11,2
      READ(LUIN,10)LNUM1,ROW,(DATA(I,J),J=1,6)
      READ(LUIN,10)LNUM2,(DATA(I+1,J),J=1,6)
      READ(LUIN,10)LNUM3,LONE
      READ(LUIN,10)LNUM4
  135 CONTINUE
      READ(LUIN,10)LNUM0,ADMIT
      DO 137 I=13,23,2
      READ(LUIN,10)LNUM1,ROW,(DATA(I,J),J=1,6)
      READ(LUIN,10)LNUM2,(DATA(I+1,J),J=1,6)
      READ(LUIN,10)LNUM3,LONE
      READ(LUIN,10)LNUM4
  137 CONTINUE
      IF(ISTAT.NE.0)PRINT,'ISTAT=',ISTAT
  140 PRINT,'ENTER THE NAME OF THE DATA FILE FOLLOWED BY A ;'
      READ,CDFIL
      CALL OPENF(LUDF,CDFIL,ISTAT,3,0,1)
      IF(ISTAT.NE.0)PRINT,'ISTAT=',ISTAT
```

```
2000        READ(LUDF,10)NPR1,NOCASE,NRC,NFOC,NCC,NTPC,NTOT,NPHSET,NSP,NSN,NEUF
2010        READ(LUDF,10)LNUM1,MCODE,RDC,T,F
2020        PI=3.14159265
2030        IF(MCODE.EQ.2)RDCT=((234.5+T)/254.5)*RDC
2040        IF(MCODE.EQ.1)RDCT=((228.+T)/248.0)*RDC
2050        RABS=RDCT*32800.
2060        FA=SQRT(F/RABS)
2070        MR=(FA*SQRT(8.0*PI))
2080        RI1=(3.0/(8.0*(MR**2)))
2090        RI2=(1.0/(MR*SQRT(2.0)))
2100        RI3=1.0+RI1+RI2
2110        RF=(FA*SQRT(PI)*RI3)
2120        RSE=(RF*RDCT)/304.8
2130        RSE1=((3.008E-4*F)/304.8)
2140        RSE=RSE+RSE1
2150        READ(LUDF,10)LNUM2,MCODE,RDC,T,RCOND,RELNG
2160        IF(MCODE.EQ.2)RDCT1=((234.5+T)/254.5)*RDC
2170        IF(MCODE.EQ.1)RDCT1=((228.0+T)/248.0)*RDC
2180        RABS1=RDCT*32800.
2190        FA1=SQRT(F/RABS1)
2200        MR1=(FA1*SQRT(8.0*PI))
2210        RI12=(3.0/(8.0*(MR1**2)))
2220        RI22=(1.0/(MR1*SQRT(2.0)))
2230        RI32=(1.0+RI12+RI22)
2240        RF1=(FA1*SQRT(PI)*RI32)
2250        RSE2=(RF1*RDCT1)/304.8
2260        RSE22=RSE2*RELNG
2270        RSE22=RSE2/RCOND
2280        RSE12 =RSE22+RSE
2290        RSE13=RSE
2300        READ(LUDF,10)LNUM3,P,GMRC,RIS,RQ,RELPER1,RELPER2,MX,RO,RI,RON,RIN,C
2310        CALL EARTH(XCCG,P,F,GMRC)
2320        XCCG=XCCG/304.8
2330        GN1=(RCOND*.7788*RIS)
2340        RCOND1=RCOND-1.0
2350        RQE=POW(RQ,RCOND1)
2360        GN2=RQE*GN1
2370        RPG=1.0/RCOND
2380        GMRCON=POW(GN2,RPG)
2390        CALL EARTH(XNNG,P,F,GMRCON)
2400        XNNG=XNNG/304.8
2410        CALL EARTH(XCNG,P,F,RQ)
2420        XCNG=XCNG/304.8
2430        CALL EARTH(XABG,P,F,CS)
2440        XABG=XABG/304.8
2450        PI=3.14159265
2460        YN=((1.0E-9)/(36.0*PI))*(RELPER1)*(PI*2.0)
2470        YD=ALOG(RO/RI)
2480        YXX=(YN/YD)
2490        YXX1=((2.0*PI)*(F))*YXX
2500        IF(MX.EQ.0)GOTO147
2510        YNN=((1.0E-9)/(36.0*PI))*(RELPER2)*(PI*2.0)
```

```
2520        YDN=ALOG(RON/RIN)
2530        YXXN=(YNN/YDN)
2540        YXX1N=((2.0*PI)*(F))*YXXN
2550    147 CONTINUE
2560        IF(MX.EQ.0)YXX1N=0.
2570        DENT(1)=RSE
2580        DENT(2)=RSE12
2590        DENT(5)=RSE1
2600        DENT(3)=XCCG
2610        DENT(4)=XNNG
2620        DENT(6)=XABG
2630        DENT(7)=XCNG
2640        DENT(8)=YXX1
2650        DENT(9)=YXX1N
2660    149 CONTINUE
2670        IF(MPMODE.NE.1)GOTO150
2680        PRINT,'ENTER RCC'
2690        READ,DENT(1)
2700        PRINT,'ENTER RNN'
2710        READ,DENT(2)
2720        PRINT,'ENTER XCC'
2730        READ,DENT(3)
2740        PRINT,'ENTER XNN'
2750        READ,DENT(4)
2760        PRINT,'ENTER RAB'
2770        READ,DENT(5)
2780        PRINT,'ENTER XAB'
2790        READ,DENT(6)
2800        PRINT,'ENTER XCN'
2810        READ,DENT(7)
2820        PRINT,'ENTER YCC'
2830        READ,DENT(8)
2840        PRINT,'ENTER YNN'
2850        READ,DENT(9)
2860    150 J=1
2870        K1=1
2880        K2=2
2890        K3=3
2900        DO 160 I=1,11,2
2910        M1=I+1
2920        DATA(I,J)=DENT(K1)
2930        DATA(M1,J)=DENT(K3)
2940        J=J+1
2950        IF(I.EQ.5)K1=K1+1
2960        IF(I.EQ.5)K3=K3+1
2970    160 CONTINUE
2980        J=2
2990        DO 170 I=1,9,2
3000        DATA(I,J)=DENT(5)
3010        IA=I+1
3020        DATA(IA,J)=DENT(6)
3030        IB=I+2
```

```
3040        JB=J-1
3050        DATA(IB,JB)=DENT(5)
3060        IC=IB+1
3070        DATA(IC,JB)=DENT(6)
3080        J=J+1
3090    170 CONTINUE
3100        J=3
3110        DO 180 I=1,7,2
3120        DATA(I,J)=DENT(5)
3130        IE=I+1
3140        DATA(IE,J)=DENT(6)
3150        JE=J-2
3160        IZ=I+4
3170        DATA(IZ,JE)=DENT(5)
3180        IF=IZ+1
3190        DATA(IF,JE)=DENT(6)
3200        J=J+1
3210    180 CONTINUE
3220        J=4
3230        DO 190 I=1,5,2
3240        DATA(I,J)=DENT(5)
3250        IE=I+1
3260        DATA(IE,J)=DENT(7)
3270        IG=I+6
3280        JF=J-3
3290        DATA(IG,JF)=DENT(5)
3300        IH=IG+1
3310        DATA(IH,JF)=DENT(7)
3320        J=J+1
3330    190 CONTINUE
3340        J=5
3350        DO 200 I=1,3,2
3360        DATA(I,J)=DENT(5)
3370        IP=I+1
3380        DATA(IP,J)=DENT(6)
3390        IQ=I+8
3400        JQ=J-4
3410        DATA(IQ,JQ)=DENT(5)
3420        IR=IQ+1
3430        DATA(IR,IQ)=DENT(6)
3440        J=J+1
3450    200 CONTINUE
3460        DATA(1,6)=DENT(5)
3470        DATA(2,6)=DENT(6)
3480        DATA(11,1)=DENT(5)
3490        DATA(12,1)=DENT(6)
3500        DO 210 I=13,24
3510        DO 205 J=1,6
3520        DATA(I,J)=0.
3530    205 CONTINUE
3540    210 CONTINUE
3550        J=1
```

```
      DO 230 I=14,18,2
      DATA(I,J)=DATA(I,J)+DENT(8)
      J=J+1
  230 CONTINUE
      MXD=4
      DO 240 I=14,18,2
      DATA(I,MXD)=DATA(I,MXD)-DENT(8)
      MXD=MXD+1
  240 CONTINUE
      MXD=1
      DO 250 I=20,24,2
      DATA(I,MXD)=DATA(I,MXD)-DENT(8)
      MXD=MXD+1
  250 CONTINUE
      MXD=4
      DO 260 I=20,24,2
      DATA(I,MXD)=DATA(I,MXD)+DENT(9)
      MXD=MXD+1
  260 CONTINUE
      PRINT,'TYPE A 1 TO BYPASS THIS CASE'
      READ,NPASS
      IF(NPASS.EQ.1)GOTO100
      IF(MPMODE.EQ.2)GOTO265
      PRINT,'ENTER THE NUMBER OF PHASE CABLES'
      READ,NPHASE
      NQC=NPHASE*2
  265 CONTINUE
      IF(MPMODE.EQ.2)NPHASE=NPHSET
      NPR1=1010
*       HERE THE COMPLEX 6X6 MATRIX IS READ FROM THE DATA FILE
      DO 280 I=1,6
      DO 270 J=1,6
      IP=(2*I)-1
      IQ=(2*I)
      RC1=DATA(IP,J)
      RC2=DATA(IQ,J)
      DATAZ(I,J)=CMPLX(RC1,RC2)
  270 CONTINUE
  280 CONTINUE
*       NOW FORM THE COMPLEX 6X6 Y MATRIX
      DO 290 I=1,6
      DO 285 J=1,6
      KP=(((2*I)-1)+12)
      KQ=((2*I)+12)
      DATAY(I,J)=CMPLX(DATA(KP,J),DATA(KQ,J))
  285 CONTINUE
  290 CONTINUE
      DO 310 I=1,6
      DO 305 J=1,6
      PRINT,DATAZ(I,J)
      PRINT,DATAY(I,J)
  305 CONTINUE
```

```
      310 CONTINUE
          IF(NPHASE.EQ.2)GOTO405
          IF(NPHASE.EQ.1)GOTO447
C           NOW FORM THE COEFFICENT COMMON TO ALL THE MODIFIERS
          Q1=(DATAZ(5,6)+DATAZ(4,4)-DATAZ(4,5)-DATAZ(4,6))
          Q2=((2*DATAZ(4,4))-(2*DATAZ(4,5)))
          Q3=((2*DATAZ(4,4))-(2*DATAZ(4,6)))
          CC=((Q2*Q3)-(Q1)**2)
C           NOW COMMON ELEMENTS IN EACH MODIFIER ARE FORMED
C           NOW FORM THE MODIFIERS
          DO 350 I=1,4
          DO 320 J=1,4
          CM1=(((DATAZ(I,5)-DATAZ(I,4))/CC)*(((DATAZ(J,6)-DAT
         &        AZ(J,4))*Q1)-(DATAZ(J,5)-DATAZ(J,4))*Q3))
          CM2=(((DATAZ(I,6)-DATAZ(I,4))/CC)*(((DATAZ(J,5)-DAT
         &        AZ(J,4))*Q1)-(DATAZ(J,6)-DATAZ(J,4))*Q2))
          DATZ3(I,J)=DATAZ(I,J)+(CM1+CM2)
      320 CONTINUE
      350 CONTINUE
C           NOW FORN THE MODIFIED Y MATRIX
          DO 370 I=1,3
          DO 355 J=1,3
          DATY2(I,J)=DATAY(I,J)
      355 CONTINUE
      370 CONTINUE
          DO 390 I=1,3
          DATY2(I,4)=DATAY(I,4)+DATAY(I,5)+DATAY(I,6)
      390 CONTINUE
          DO 400 K=1,3
          DATY2(4,K)=DATAY(K,4)+DATAY(K,5)+DATAY(K,6)
      400 CONTINUE
          DATY2(4,4)=(DATAY(4,4)+(2*DATAY(4,5))+(2*DATAY(4,6))+DATAY(5,5)
         &      +(2*DATAY(5,6))+DATAY(6,6))
      405 NPM=NPHASE+1
          DO 410 I=1,2
          DATZ2(I,3)=DATAZ(I,4)
          DATZ2(3,I)=DATAZ(4,I)
      410 CONTINUE
          DATZ2(3,4)=DATAZ(4,5)
          DO 420 J=1,2
          DATZ2(J,4)=DATAZ(J,5)
          DATZ2(4,J)=DATAZ(5,J)
      420 CONTINUE
          DATZ2(4,3)=DATAZ(5,4)
          DATZ2(3,3)=DATAZ(4,4)
          DATZ2(4,4)=DATAZ(5,5)
          DO 430 I=1,2
          DATZ2(I,I)=DATAZ(I,I)
      430 CONTINUE
          DATZ2(1,2)=DATAZ(1,2)
          DATZ2(2,1)=DATAZ(2,1)
          IF(NPHASE.NE.2)GOTO447
```

```
      DO 435 I=1,NPM
      DO 432 J=1,NPM
      CM3=((DATZ2(I,4)-DATZ2(I,3))*(DATZ2(J,3)-DATZ2(J,4)))
      CM4=(((2*DATZ2(3,4))-(DATZ2(3,3))-(DATZ2(4,4))))
      ZMOD5=-(CM3/CM4)
      DATZ3 I,J)=DATZ2(I,J)+ZMOD5
  432 CONTINUE
  435 CONTINUE
      DATY2(1,1)=DATAY(1,1)
      DATY2(1,2)=DATAY(1,2)
      DATY2(2,1)=DATAY(2,1)
      DATY2(2,2)=DATAY(2,2)
      DATY2(1,3)=DATAY(1,4)+DATAY(1,5)
      DATY2(2,3)=DATAY(2,4)+DATAY(2,5)
      DO 445 I=1,2
      DATY2(3,I)=DATAY(4,I)+DATAY(5,I)
  445 CONTINUE
      DATY2(3,3)=(DATAY(4,4)+DATAY(5,5)+(2*DATAY(4,5)))
*       NOW WRITE THE OUTPUT FILE
  447 WRITE(LUOU,20)NPR1,NOCASE,NRC,NFQC,NCC,NTPC,NTOT,NPHSET,NSP,
     &                NSN,NEUF,F/1000.
      IF(NPHASE.EQ.2)GOTO465
      NPM=NPHASE+1
      IF(NPHASE.EQ.1)DATZ3(1,1)=DATAZ(1,1)
      IF(NPHASE.EQ.1)DATZ3(2,1)=DATAZ(4,1)
      IF(NPHASE.EQ.1)DATZ3(1,2)=DATAZ(1,4)
      IF(NPHASE.EQ.1)DATZ3(2,2)=DATAZ(4,4)
      IF(NPHASE.EQ.1)DATY2(1,1)=DATAY(1,1)
      IF(NPHASE.EQ.1)DATY2(1,2)=DATAY(1,4)
      IF(NPHASE.EQ.1)DATY2(2,1)=DATAY(4,1)
      IF(NPHASE.EQ.1)DATY2(2,2)=DATAY(4,4)
      DO 462 K=1,NPM
      DO 448 L=1,NPM
      NPR1=NPR1+10
      WRITE(LUOU,30)NPR1,DATZ3(K,L)
  448 CONTINUE
  462 CONTINUE
  465 NS=1
      IF(NPHASE.NE.2)GOTO470
      DO 467 I=1,NPM
      DO 466 J=1,NPM
      NPR1=NPR1+10
      WRITE(LUOU,30)NPR1,DATZ3(I,J)
  466 CONTINUE
  467 CONTINUE
  470 NT=1
      DO 520 K=1,NPM
      DO 510 L=1,NPM
      NPR1=NPR1+10
      WRITE(LUOU,30)NPR1,DATY2(K,L)
  510 CONTINUE
  520 CONTINUE
```

```
5120        WRITE(LUOU,40)NFN
5130        PRINT,'TYPE A 1 TO PROCESS A NEW INPUT FILE'
5140        READ,I
5150        CALL DETACH(LUIN,ISTAT,)
5160        CALL DETACH(LUOU,ISTAT,)
5170        CALL DETACH(LUDF,ISTAT,)
5180        IF(I.EQ.1)GOTO100
5190    999 PRINT,'STOP AT STATEMENT #999'
5200        STOP
5210        END
5220        SUBROUTINE EARTH(XS,PS,FS,GS)
5230        G1=SQRT(PS/FS)
5240        GF=(ALOG10(G1/GS))
5250        XS=(8.8193E-4)*(GF)*FS
5260        XS=XS+(3.8929E-3*FS)
5270        RETURN
5280        END
```

```
0010** RUN *=ZPTRANS1(ULID,CORE=30K,NWARN)LIBRARY4:LIBRARY/ZFTA,K
0020 ************************************************************
0030 *      THIS PROGRAM ANALYZES DISTRIBUTION TRANSFORMERS      *
0040 *      USING THE LUMPED PARAMETER MODEL.  THE USER MAY ENTER *
0050 *      VALUES FOR THE VARIOUS CIRCUIT ELEMENTS OR MAY CHOOSE *
0060 *      TO REMOVE THOSE ELEMENTS FROM THE ANALYSIS SIMPLY BY  *
0070 *      ENTERING A VALUE OF ZERO WHEN PROMPTED BY THE PROGRAM. *
0080 *      ANALYSIS MAY BE IN ONE OF TWO MODES:                  *
0090 *                                                            *
0100 *          MODE 1:  PRIMARY       --->    SECONDARY         *
0110 *          MODE 2:  SECONDARY     --->    PRIMARY           *
0120 *                                                            *
0130 *      COMPUTATION RESULTS IN VALUES FOR THE MAGNITUDES OF THE *
0140 *      INPUT ADMITTANCE AND THE VOLTAGE TRANSFER RATIO, BOTH *
0150 *      EXPRESSED IN DB, AS WELL AS THEIR RELATIVE PHASE ANGLES, *
0160 *      IN DEGREES.  A PLOT FILE IS GENERATED WHICH CONSISTS OF *
0170 *      TWO PLOTS.  THE FIRST IS OF THE VOLTAGE TRANSFER RATIO, *
0180 *      ITS MAGNITUDE IN DB AND ITS PHASE ANGLE IN DEGREES.   *
0190 *      THE SECOND IS OF THE INPUT ADMITTANCE, ALSO WITH MAG- *
0200 *      NITUDE IN DB AND PHASE ANGLE IN DEGREES.              *
0210 *          CODING FOR OTHER DEPENDANT VARIABLE COMBINATIONS   *
0220 *      COULD EASILY BE ADDED.  THE RANGE OF FREQUENCIES      *
0230 *      ENCOMPASSED BY THIS ANALYSIS IS FROM 2 K-HZ TO 100 K-HZ, *
0240 *      IN 500 HZ INCREMENTS.                                 *
0250 ************************************************************
0260
0270 *            STATEMENTS NUMBERED 10 THROUGH 49 ARE OUTPUT FORMATS
0280
0290     10 FORMAT('THIS PROGRAM USES THE LUMPED PARAMETER MODEL'/
0300       &       'TO ANALIZE DISTRIBUTION TRANSFORMERS')
0310     11 FORMAT(/'MODE 1 CALCULATES PRIMARY  ---> SECONDARY'/
0320       &        'MODE 2 CALCULATES SECONDARY --> PRIMARY'/)
0330     12 FORMAT(/'ENTER 1 FOR MODE 1 OR 2 FOR MODE 2')
0340     13 FORMAT(//'LOAD INDUCTANCE  =',F15.6,' MILLIHENRYS')
0350     14 FORMAT('LOAD RESISTANCE  =',F15.6,' OHMS')
0360     15 FORMAT('ENTER CR IF OKAY, ELSE ENTER 1')
0370     16 FORMAT(/'ENTER LOAD INDUCTANCE   (IN MILLIHENRYS)')
0380     17 FORMAT(/'ENTER LOAD RESISTANCE   (IN OHMS)')
0390     18 FORMAT(//'FEEDER CAPACITANCE =',F15.6,' MICROFARADS')
0400     19 FORMAT('FEEDER CONDUCTANCE =',F15.6,' MHOS')
0410     20 FORMAT(/'ENTER FEEDER CAPACITANCE   (IN MICROFARADS)')
0420     21 FORMAT(/'ENTER FEEDER CONDUCTANCE   (IN MHOS)')
0430     22 FORMAT(//'WINDING RESISTANCE =',F15.6,' OHMS          REFERRED TO')
0440     23 FORMAT('WINDING INDUCTANCE =',F15.6,' MILLIHENRYS   SECONDARY')
0450     24 FORMAT(/'ENTER WINDING RESISTANCE   (IN OHMS)')
0460     25 FORMAT(/'ENTER WINDING INDUCTANCE   (IN MILLIHENRYS)')
0470     26 FORMAT(//'TURNS RATIO OF IDEAL TRANSFOPMER =',F15.6)
0480     27 FORMAT('STRAY CAPACITANCE                  =',F15.6,' MICROFARADS')
0490     28 FORMAT(/'ENTER TURNS RATIO')
```

```
0500     29 FORMAT(/'ENTER STRAY CAPACITANCE    (IN MICROFARADS)')
0510     30 FORMAT(//'CORE INDUCTANCE  =',F15.6,' MILLIHENRYS')
0520     31 FORMAT('CORE CONDUCTANCE =',F15.6,' MHOS')
0530     32 FORMAT(/'ENTER CORE INDUCTANCE    (IN MILLIHENRYS)')
0540     33 FORMAT(/'ENTER CORE CONDUCTANCE    (IN MHOS)')
0550     34 FORMAT(/'DO YOU WANT OUTPUT AT THE TERMINAL ?')
0560     35 FORMAT(/'OUTPUT FILE =',A9)
0570     36 FORMAT('COULD NOT DETACH FILE',A10,' WITH STATUS =',I3)
0580     37 FORMAT(A8,A1)
0590     38 FORMAT('OPENED FILE =',A10)
0600     39 FORMAT('MODE OF OPERATION =',I3)
0610     40 FORMAT(///'FREQ. (HZ)',T14,'VTR (IN DB)',T30,'VTR PHASE',
0620       &       T44,'YINPUT (DB)',T58,'YINPUT PHASE'//)
0630     41 FORMAT(F9.1,T13,1PE13.6,T28,E13.6,T43,E13.6,2X,E13.6)
0640     42 FORMAT(/'WROTE OUTPUT FILE =',A10)
0650     43 FORMAT(///'ENTER A VALUE OF  0.  FOR ELEMENTS YOU WISH TO IGNORE')
0660     44 FORMAT(//'BASE FREQUENCY FOR FREQUENCY DEPENDANT',
0670       &       'WINDING RESISTANCE =',F15.6,' HERTZ')
0680     45 FORMAT(/'ENTER BASE FREQUENCY    (IN HERTZ)')
0690     46 FORMAT(/'ENTER NAME OF OUTPUT FILE')
0700     47 FORMAT(//'RANGE OF VTR MAGNITUDE     =',F6.1,' TO ',F6.1,' DB')
0710     48 FORMAT('RANGE OF VTR PHASE ANGLES =',F6.1,' TO ',F6.1,' DEGREES')
0720     49 FORMAT('RANGE OF YINPUT MAGNITUDE =',F6.1,' TO ',F6.1,' DB')
0730     50 FORMAT(/'ENTER LOWER AND UPPER VTR MAGNITUDE LIMITS'
0740       &       '(IN DB), SEPERATED BY A COMMA')
0750     51 FORMAT(/'ENTER LOWER AND UPPER VTR PHASE ANGLE LIMITS'
0760       &       '(IN DEGREES), SEPERATED BY A COMMA')
0770     52 FORMAT(/'ENTER LOWER AND UPPER YINPUT MAGNITUDE LIMITS'
0780       &       '(IN DB), SEPERATED BY A COMMA')
0790     53 FORMAT(/'DO YOU WISH TO ANALIZE ANOTHER TRANSFORMER ?')
0800     54 FORMAT(/'FOR OPEN CIRCUIT SECONDARY, RESET BOTH THE LOAD'
0810       &       'INDUCTANCE AND LOAD RESISTANCE TO ZERO')
0820     55 FORMAT('RANGE OF YINPUT PHASE     =',F6.1,' TO ',F6.1,' DEGREES')
0830     56 FORMAT(/'ENTER LOWER AND UPPER YINPUT PHASE LIMITS',
0840       &       '(IN DEGREES), SEPERATED BY A COMMA')
0850
0860 *            VARIABLE DECLARATIONS
0870
0880        IMPLICIT COMPLEX(Y,Z)
0890        COMPLEX VTR(200),ONE,VT
0900        CHARACTER OUFILE*9/'DTOUT1;'/,FNAME*8,NAME*6/'ZPTRAN'/
0910        CHARACTER*30 IXC,IYC,ITI
0920        REAL LW,RW,LC,GC,PL,LL,CF,GF,C,N,TWOPI,FREQ
0930        DIMENSION ADPLOT(200,2),VTRPLT(200,2),PH1PLT(200,2),PH2PLT(200,2)
0940        DATA BASE/20000.0/,NP/197/,NDIM/200/,MDIM/2/,MV/1/,
0950       &     TWOPI/6.28318/,NXC/18/,X0/1.0/,X1/0.0/,X2/100000.00/,
0960       &     ONE/(1.0,0.0)/,XS/6793.48/,XY0/0.51/,XYD/9.84/,
0970       &     LW/0.086/,
0980       &     RW/0.14/,
0990       &     LC/0.0/,
1000       &     GC/0.0/,
1010       &     RL/10/,
```

```
1020       &      LL/0.0/,
1030       &      CF/0.0/,
1040       &      RF/0.1/,
1050       &      CL/0.01/,
1060       &       /21.75/,
1070       &      IOUT/3/,
1080       &      VT21/-50.00/,
1090       &      /TR2/20.00/,
1100       &      PH1/-180.00/,
1110       &      PH2/180.00/,
1120       &      PH3/-180.00/,
1130       &      PH4/180.00/,
1140       &      AC1/-50.00/,
1150       &      AC2/20.00/,
1160       &      DEG/57.29578/,
1170       &      IXC/'FREQUENCY IN HERTZ'/
1180
1190 *             USE AN ARITHMETIC STATEMENT FUNCTION TO FIND
1200 *             THE FREQUENCY DEPENDANT MISSING RESISTANCE.  USE
1210 *             THE VALUES ENTERED FOR RW AND THE BASE FREQUENCY
1220 *             TO DETERMINE VALUES AT OTHER FREQUENCIES.
1230 *             ALSO USE ARITHMETIC STATEMENT FUNCTIONS TO CALCU-
1240 *             LATE MAGNITUDES IN DB FOR THE VTR AND INPUT ADMITTANCE
1250 *             AS WELL AS THE PHASE ANGLE OF THE VOLTAGE TRANSFER RATIO.
1260
1270       REF(X) = (X / BASE)**1.4
1280       DBF(Y) = 20.0 * ALOG10(CABS(Y))
1290       PHF(Y) = DEG * ATAN2(AIMAG(Y),REAL(Y))
1300
1310 *             NOW DETERMINE WHICH MODE OF OPERATION THE USER
1320 *             DESIRES AND READ IN THE APPROPRIATE VARIABLES
1330 *
1340 *             MODE 1 CALCULATES PRIMARY  ---> SECONDARY
1350 *             MODE 2 CALCULATES SECONDARY --> PRIMARY
1360
1370       PRINT 10
1380       PRINT 11
1390   100 LUOUT = 20
1400       PRINT 12
1410       READ,MODE
1420       IF(MODE.EQ.1.OR.MODE.EQ.2) PRINT 43
1430       IF(MODE.NE.1) GOTO 130
1440
1450 *             READ IN LOAD INDUCTANCE AND CONDUCTANCE FOR MODE 1
1460 *             OPERATION AND SET FEEDER CAPACITANCE AND CONDUCTANCE
1470 *             TO ZERO
1480
1490       PRINT 54
1500   110 PRINT 13,LL
1510       PRINT 14,RL
1520       PRINT 15
1530       READ,IOKAY
```

```
1540        IF(IOKAY.EQ.0) GOTO 120
1550        PRINT 16
1560        READ,LL
1570        PRINT 17
1580        READ,RL
1590        GOTO 110
1600
1610    120 CF = 0.0
1620        GF = 0.0
1630        GOTO 150
1640
1650  *             READ IN FEEDER CAPACITANCE AND CONDUCTANCE FOR MODE 2
1660  *             OPERATION AND SET LOAD INDUCTANCE AND CONDUCTANCE
1670  *             TO ZERO
1680
1690    130 IF(MODE.NE.2) GOTO 100
1700        PRINT 18,CF
1710        PRINT 19,GF
1720        PRINT 15
1730        READ,IOKAY
1740        IF(IOKAY.EQ.0) GOTO 140
1750        PRINT 20
1760        READ,CF
1770        PRINT 21
1780        READ,GF
1790        GOTO 130
1800
1810    140 LL = 0.0
1820        RL = 0.0
1830
1840  *             READ IN ALL DATA THAT IS USED FOR BOTH MODE 1 AND
1850  *             MODE 2 OPERATION
1860
1870    150 PRINT 44,BASE
1880        PRINT 15
1890        READ,IOKAY
1900        IF(IOKAY.EQ.0) GOTO 160
1910        PRINT 45
1920        READ,BASE
1930        GOTO 150
1940
1950    160 PRINT 22,RN
1960        PRINT 23,LN
1970        PRINT 15
1980        READ,IOKAY
1990        IF(IOKAY.EQ.0) GOTO 170
2000        PRINT 24
2010        READ,RW
2020        PRINT 25
2030        READ,LW
2040        GOTO 160
2050
```

```
2060    170 PRINT 26,N
2070        PRINT 27,C
2080        PRINT 15
2090        READ,IOKAY
2100        IF(IOKAY.EQ.0) GOTO 180
2110        PRINT 28
2120        READ,N
2130        PRINT 29
2140        READ,C
2150        GOTO 170
2160
2170    180 PRINT 30,LC
2180        PRINT 31,GC
2190        PRINT 15
2200        READ,IOKAY
2210        IF(IOKAY.EQ.0) GOTO 190
2220        PRINT 32
2230        READ,LC
2240        PRINT 33
2250        READ,GC
2260        GOTO 180
2270
2280 *              NOW DETERMINE THE DESTINATION OF THE PROGRAM OUTPUT.
2290 *              DEFAULT DESTINATION IS A FILE NAMED 'DTOUT1' BUT THE
2300 *              USER CAN SPECIFY OUTPUT TO A DIFFERENT FILE OR TO THE
2310 *              TERMINAL.  IN ANY CASE, THE PLOT FILE IS WRITTEN.
2320
2330    190 PRINT 35,OUFILE
2340        PRINT 15
2350        READ,IOKAY
2360        IF(IOKAY.EQ.0) GOTO 210
2370        IF(IOUT.EQ.0) GOTO 200
2380        CALL DETACH(LUOUT,ISTAT,)
2390        IF(ISTAT.NE.0) PRINT 36,OUFILE,ISTAT
2400    200 PRINT 34
2410        PRINT 15
2420        READ,IOKAY
2430        IF(IOKAY.EQ.0) GOTO 220
2440        PRINT 46
2450        READ,FNAME
2460        ENCODE(OUFILE,37)FNAME,';'
2470        IOUT = 0
2480        GOTO 190
2490
2500    210 CONTINUE
2510        IF(IOUT.EQ.1) GOTO 230
2520        CALL OPENF(LUOUT,OUFILE,ISTAT)
2530        IF(ISTAT.NE.0) CALL ERRSTT(NAME,210,LUOUT,ISTAT,0)
2540        PRINT 38,OUFILE
2550        IOUT = 1
2560        GOTO 230
2570
```

```
2580    220 LOOUT = 6
2590
2600 *              OVERRIDE THE RANGE OF VALUES TO BE USED ON
2610 *             THE ORDINATES OF THE FOUR PLOTS.
2620
2630    230 PRINT 47,VTR1,VTR2
2640        PRINT 48,PH1,PH2
2650        PRINT 49,AD1,AD2
2660        PRINT 55,PH3,PH4
2670        PRINT 15
2680        READ,IOKAY
2690        IF(IOKAY.EQ.0) GOTO 240
2700        PRINT 50
2710        READ,VTR1,VTR2
2720        PRINT 51
2730        READ,PH1,PH2
2740        PRINT 52
2750        READ,AD1,AD2
2760        PRINT 56
2770        READ,PH3,PH4
2780        GOTO 230
2790
2800 *              NOW THAT ALL INPUT DATA HAS BEEN READ IN, CONVERT
2810 *              MILLIHENRYS TO HENRYS AND MICROFARADS TO FARADS.
2820 *              THIS WILL EASE COMPUTATIONS WHEN CONVERSIONS TO
2830 *              IMPEDANCES AND ADMITTANCES ARE REQUIRED
2840
2850    240 C  = C  * 1.0E-06
2860        CF = CF * 1.0E-06
2870        LC = LC * 1.0E-03
2880        LL = LL * 1.0E-03
2890        LW = LW * 1.0E-03
2900
2910 *              CODE EXECUTED BY MODE 1 OPERATION IS SEPERATED
2920 *              FROM THAT EXECUTED BY MODE 2 OPERATION TO FACILITATE
2930 *              EASY COMPREHENSION OF THE PROGRAM LOGIC
2940
2950        FREQ = 1500.00
2960        IF(MODE.NE.1) GOTO 270
2970        DO 260 I=1,197
2980        FREQ  = FREQ + 500.0
2990        W     = TWOPI * FREQ
3000        A     = W * C
3010        IF(LC.NE.0.0) A = A-1.0/(W*LC)
3020        YINP  = CMPLX(GC,A)
3030        VT    = ONE / N
3040        IF(RL.EQ.0.0.AND.LL.EQ.0.0) GOTO 250
3050        ZLOAD = CMPLX(RL,W*LL)
3060        ZWIND = CMPLX(REF(FREQ),W*LW)
3070        YCOMB = ONE / (ZLOAD + ZWIND)
3080        VT    = (YCOMB * ZLOAD) / N
3090        YINP  = YINP + YCOMB/(N*N)
```

```
3100    250 CONTINUE
3110        VTR(I)       = VT
3120        VTRPLT(I,1) = FREQ
3130        VTRPLT(I,2) = DBF(VT)
3140        PH1PLT(I,1) = FREQ
3150        PH1PLT(I,2) = PHF(VT)
3160        ADPLOT(I,1) = FREQ
3170        ADPLOT(I,2) = DBF(YINP)
3180        PH2PLT(I,1) = FREQ
3190        PH2PLT(I,2) = PHF(YINP)
3200    260 CONTINUE
3210        GOTO 290
3220
3230
3240
3250
3260 *              EXECUTE THE FOLLOWING CODE FOR MODE 2 OPERATION
3270
3280    270 CONTINUE
3290        DO 280 I=1,197
3300        FREQ         = FREQ + 500.00
3310        W            = TWOPI * FREQ
3320        A            = W * (CF + C)
3330        IF(LC.NE.0.0) A = A-1.0/(W*LC)
3340        YINP         = CMPLX(GF + GC,A)
3350        VT           = N / (CMPLX(GN,W*N)*YINP*CMPLX(REF(FREQ),W*LW))
3360        YINP         = N * VT * YINP
3370        VTR(I)       = VT
3380        VTRPLT(I,1) = FREQ
3390        VTRPLT(I,2) = DBF(VT)
3400        PH1PLT(I,1) = FREQ
3410        PH1PLT(I,2) = PHF(VT)
3420        ADPLOT(I,1) = FREQ
3430        ADPLOT(I,2) = DBF(YINP)
3440        PH2PLT(I,1) = FREQ
3450        PH2PLT(I,2) = PHF(YINP)
3460    280 CONTINUE
3470
3480 *              NOW THAT CALCULATIONS ARE COMPLETE, RECONVERT
3490 *              HENRYS TO MILLIHENRYS AND FARADS TO MICRO-
3500 *              FARADS.  THIS WILL EASE USER OPERATION.
3510
3520    290 C  = C  * 1.0E 06
3530        CF = CF * 1.0E 06
3540        LC = LC * 1.0E 03
3550        LL = LL * 1.0E 03
3560        LW = LW * 1.0E 03
3570
3580 *              NOW BEGIN OUTPUT SECTION OF THE PROGRAM.
3590
3600        WRITE(LUOUT,39)MODE
3610        WRITE(LUOUT,44)BASE
```

```
3620        IF(MODE.EQ.1) WRITE(LUOUT,13)LL
3630        IF(MODE.EQ.1) WRITE(LUOUT,14)PL
3640        IF(MODE.EQ.2) WRITE(LUOUT,13)CF
3650        IF(MODE.EQ.2) WRITE(LUOUT,14)GF
3660        WRITE(LUOUT,22)RM
3670        WRITE(LUOUT,23)LM
3680        WRITE(LUOUT,26)N
3690        WRITE(LUOUT,27)C
3700        WRITE(LUOUT,30)LC
3710        WRITE(LUOUT,31)GC
3720        WRITE(LUOUT,40)
3730        DO 300 I=1,197
3740        WRITE(LUOUT,41)VTRPLT(I,1),VTRPLT(I,2),PH1PLT(I,2),
3750       &                ADPLOT(I,2),PH2PLT(I,2)
3760    300 CONTINUE
3770        IF(LUOUT.NE.6) PRINT 42,OUFILE
3780
3790 *              SET ARGUMENTS FOR VTR PLOT AND CALL PLOTTING ROUTINE
3800
3810        ID    = ID + 1
3820        IYC   = 'VTR MAGNITUDE IN DB'
3830        ITI   = 'VOLTAGE TRANSFER RATIO'
3840        NYC   = 19
3850        NTI   = 22
3860        IFO   = 1
3870        PRINT,'PLOTTING VTR           VS. FREQUENCY'
3880        CALL ZPLOTT(ID,NP,MV,VTRPLT,NDIM,MDIM,IXC,IYC,ITI,XO,
3890       &            X1,X2,XS,IFO,NXC,NYC,NTI,XYO,VTR1,VTR2,XYD)
3900
3910 *              SET ARGUMENTS FOR VTR PHASE PLOT AND CALL PLOTTING ROUTINE
3920 *              TO PUT BOTH DB & PHASE ON THE SAME SHEET
3930
3940        IYC   = 'PHASE ANGLE IN DEGREES'
3950        NYC   = 22
3960        IFO   = 2
3970        PRINT,'PLOTTING VTR PHASE     VS. FREQUENCY'
3980        CALL ZPLOTT(ID,NP,MV,PH1PLT,NDIM,MDIM,IXC,IYC,ITI,XO,
3990       &            X1,X2,XS,IFO,NXC,NYC,NTI,XYO,PH1,PH2,XYD)
4000
4010 *              SET ARGUMENTS FOR INPUT ADMITTANCE PLOT AND CALL
4020 *              PLOTTING ROUTINE.
4030
4040        IYC   = 'INPUT ADMITTANCE IN DB'
4050        ITI   = 'INPUT ADMITTANCE VS. FREQUENCY'
4060        NYC   = 22
4070        NTI   = 30
4080        IFO   = 1
4090        PRINT,'PLOTTING YINPUT        VS. FREQUENCY'
4100        CALL ZPLOTT(ID,NP,MV,ADPLOT,NDIM,MDIM,IXC,IYC,ITI,XO,
4110       &            X1,X2,XS,IFO,NXC,NYC,NTI,XYO,AD1,AD2,XYD)
4120
4130 *              SET ARGUMENTS FOR INPUT ADMITTANCE PHASE VS.
```

```
4140 *            FREQUENCY PLOT ... CALL PLOTTING ROUTINE
4150
4160       IYC  = 'PHASE ANGLE IN DEGREES'
4170       NYC  = 22
4180       IFO  = 1
4190       PRINT,'PLOTTING YINPUT PHASE VS. FREQUENCY'
4200       CALL ZPLOTT(ID,NP,NV,PH2PLT,NDIM,MDIM,IXC,IYC,ITI,X0,
4210      &            X1,X2,XS,IFO,NXC,NYC,NTI,XY0,PH3,PH4,XYD)
4220
4230 *            NOW SEE IF THE USER WISHES TO EXECUTE THE PROGRAM AGAIN
4240
4250       PRINT 53
4260       PRINT 15
4270       READ,IOKAY
4280       IF(IOKAY.EQ.0) GOTO 100
4290       STOP
4300
4310 *            DEFINITION OF VARIABLES IN 'ZPTRANS1'
4320
4330 *            A       - TEMPORARY COMPLEX VARIABLE
4340 *            ADPLOT  - ARRAY WITH YINPUT VS. FREQUENCY PLOT INFORMATION
4350 *            AD1     - LOWER LIMIT OF ORDINATE FOR YINPUT PLOT
4360 *            AD2     - UPPER LIMIT OF ORDINATE FOR YINPUT PLOT
4370 *            BASE    - BASE FREQ. FOR FREQUENCY DEPENDANT RESISTANCES
4380 *            C       - STRAY CAPACITANCE
4390 *            CF      - FEEDER CAPACITANCE
4400 *            DEG     - CONVERSION FACTOR FROM RADIANS TO DEGREES
4410 *            FNAME   - NAME TO BE GIVEN OUTPUT FILE
4420 *            FREQ    - FREQUENCY (IN HERTZ)
4430 *            GC      - CORE CONDUCTANCE
4440 *            GF      - FEEDER CONDUCTANCE
4450 *            ID      - IDENTIFICATION NUMBER OF PLOT FILE
4460 *            IFO     - PLOTTER CONTROL CODE
4470 *            ITI     - TITLE OF PLOT
4480 *            IXC     - ABCISSA LABEL FOR PLOT
4490 *            IYC     - ORDINATE LABEL FOR PLOT
4500 *            IOUT    - 'FILE BY THAT NAME EXISTS' FLAG
4510 *            IOKAY   - 'VALUES ACCEPTABLE' RESPONSE
4520 *            ISTAT   - STATUS RETURNED BY SUBROUTINE 'OPENF'
4530 *            LC      - CORE INDUCTANCE
4540 *            LL      - LOAD INDUCTANCE
4550 *            LUOUT   - OUTPUT DEVICE CODE
4560 *            LW      - WINDING INDUCTANCE - REFERRED TO SECONDARY
4570 *            MDIM    - COLUMN DIMENSION OF VTRPLT,PH1PLT,PH2PLT, OR ADPLOT
4580 *            MODE    - MODE OF OPERATION (1 OR 2)
4590 *            NV      - NUMBER OF DEPENDANT VARIABLES TO BE PLOTTED
4600 *            N       - TURNS RATIO OF IDEAL TRANSFORMER
4610 *            NAME    - THE NAME OF THIS PROGRAM (FOR ERROR MESSAGES)
4620 *            NDIM    - ROW DIMENSION OF VTRPLT,PH1PLT,PH2PLT, OR ADPLOT
4630 *            NP      - NUMBER OF POINTS ON ABCISSA OF PLOT
4640 *            NTI     - NUMBER OF CHAPACTERS IN ITI
4650 *            NXC     - NUMBER OF CHARACTERS IN IXC
```

```
4660 *              IYC     - NUMBER OF CHARACTERS IN IYC
4670 *              OUFILE  - OUTPUT FILE NAME
4680 *              PH1PLT  - ARRAY WITH VTR PHASE VS. FREQUENCY PLOT INFORMATION
4690 *              PH2PLT  - ARRAY WITH YINPUT PHASE PLOT INFORMATION
4700 *              PH1     - LOWER LIMIT OF ORDINATE FOR VTR PHASE ANGLE PLOT
4710 *              PH2     - UPPER LIMIT OF ORDINATE FOR VTR PHASE ANGLE PLOT
4720 *              PH3     - LOWER LIMIT OF ORDINATE FOR YINPUT PHASE PLOT
4730 *              PH4     - UPPER LIMIT OF ORDINATE FOR YINPUT PHASE PLOT
4740 *              RL      - LOAD RESISTANCE
4750 *              RW      - WINDING RESISTANCE (FREQUENCY DEPENDANT)
4760 *                        REFERRED TO SECONDARY WHEN IT IS ENTERED
4770 *              TWOPI   - TWO TIMES PI
4780 *              VTR     - VOLTAGE TRANSFER RATIO
4790 *              VTRPLT  - VTR MAGNITUDE VS. FREQUENCY PLOT INFORMATION
4800 *              VTR1    - LOWER LIMIT OF ORDINATE FOR VTR PLOT
4810 *              VTR2    - UPPER LIMIT OF ORDINATE FOR VTR PLOT
4820 *              X0      - ABCISSA COORDINATE OF ORIGIN FOR PLOT
4830 *              X1      - FIRST VALUE ON ABCISSA OF PLOT
4840 *              X2      - LAST VALUE ON ABCISSA OF PLOT
4850 *              XS      - ABCISSA SCALE IN UNITS PER INCH
4860 *              XY0     - ORDINATE COORDINATE OF ORIGIN FOR PLOT
4870 *              XYD     - DISTANCE IN INCHES TO BE SPANNED BY XY2-XY1
4880 *              YCOMB   - ADMITTANCE OF LOAD & WINDING COMBINED
4890 *              ZLOAD   - IMPEDANCE OF THE REFERRED LOAD
4900 *              ZWIND   - IMPEDANCE OF THE WINDING - REFERRED TO SECONDARY
4910
4920        END
```

2. SUBROUTINE AND RELATED LISTINGS

NOTE: ON THE FOLLOWING LISTING OF SUBRLIST, SUBROUTINES MARKED BY A ● ARE NOT CURRENTLY USED IN THE DIFNAP SYSTEM PROGRAMS.

LIST OF SUBROUTINES
* NOT (INTENTIONALLY) CONTAINED IN R.RUSTAY USER LIBRARY

ABCDEM	COMPUTES DIAGONAL FORM OF ABCD MATRICES FOR LINE LENGTH X
ABCDEN	COMPUTES NXN MATRIX FORM OF ABCD MATRICES FOR LINE LENGTH X
ADBMPY	COMPUTES SPECIAL MATRIX PRODUCT C=A*D*B WHERE D IS DIAGONAL
CCMMPY	COMPUTES THE COMPLEX CHAIN MATRIX MULTIPLY
CDAPBC	COMPUTES SPECIAL MATRIX OPERATION W=XD*YD*Z, XD,YD DIAGONAL
CHYPER	COMPUTES THE COMPLEX HYPERBOLIC SINH,COSH OF COMPLEX ARGUMENT
CMAMBC	COMPUTES SPECIAL MATRIX OPERATION D=A-B*C
CMAPBC	COMPUTES SPECIAL MATRIX OPERATION D=A+B*C
CMDBAN	PRINTS OR WRITES FOR DIAGNOSIS A COMPLEX MATRIX IN DB-ANGLE FORM
CMINRF	COMPUTES AND IMPROVES (BY FADDEEVA) COMPLEX MATRIX INVERSE
CMTABC	COMPUTES COMPLEX MATRIX MULTIPLY C=ATRANSPOSE*BCONJUGATE
CMTADD	COMPUTES MATRIX ADD C=A+B
CMTCMP	COMPARES AN INPUT COMPLEX MATRIX WITH STANDARD MATRIX
CMTCOP	COMPUTES MATRIX COPY B=A WITH A AND B EACH SQUARE
CMTCPY	COMPUTES NONSQUARE MATRIX COPY B=A AND ZEROES BORDER OF B
CMTCRC	INTERCHANGES TWO SPECIFIED ROWS AND COLUMNS AFTER COPYING
CMTDWD	COMPUTES SPECIAL MATRIX OPERATION DW=AD*BW*CD, AD,CD DIAGONAL
CMTIDE	PERFORMS DIAGNOSTIC CHECK OF MATRIX FOR IDENTITY OR DIAGONAL
CMTIRC	INTERCHANGES TWO SPECIFIED ROWS AND COLUMNS IN PLACE
CMTMPT	COMPUTES MATRIX TRANSPOSE MULTIPLY C=A*BT
CMTMPY	COMPUTES MATRIX MULTIPLY C=A*B
CMTPOL	PRINTS OR WRITES FOR DIAGNOSIS A COMPLEX MATRIX IN POLAR FORM
CMTPRT	PRINTS OR WRITES FOR DIAGNOSIS A COMPLEX MATRIX
CMTRAN	COMPUTES MATRIX TRANSPOSE OPERATION B=AT AND ZEROES BORDER OF B
CMTSUB	COMPUTES MATRIX SUBTRACT C=A-B
CMTTMP	COMPUTES MATRIX TRANSPOSE MULTIPLY C=AT*B
CMTZRO	ZEROES BOARDER OF NONSQUARE MATRIX A
CMVMPY	COMPUTES THE MATRIX-VECTOR MULTIPLY W=A*V
CSYADJ	FORCES EXACT SYMMETRY ON A SPECIFIED NXN MATRIX
CVDBAN	PRINTS OR WRITES FOR DIAGNOSIS A COMPLEX VECTOR IN DB-ANGLE FORM
CVECMP	COMPARES AN INPUT COMPLEX VECTOR WITH A STANDARD VECTOR
CVEPOL	PRINTS OR WRITES FOR DIAGNOSIS A COMPLEX VECTOR IN POLAR FORM
CVEPRT	PRINTS OR WRITES FOR DIAGNOSIS A COMPLEX VECTOR
CZYRED	COMPUTES REDUCED Z,Y MATRICES FOR ZERO POTENTIAL ASSUMPTIONS
DPUFRW	SPECIAL ROUTINE FOR READ/WRITE OF FILE DPULIJKN
ERRSTP	GENERAL STOP ROUTINE WITH NAME AND CODE CAPABILITY
ERRSTT	GENERAL STOP ROUTINE WITH NAME, CODE AND INTEGER OUTPUT
FILSTP	PROVIDES FILE READ ERROR IDENTIFICATION AND ABORT
FORMFE	PRODUCES A FORMFEED TO TERMINET 1200 WITH SUITABLE TIMING DELAYS
LINEPU	COMPUTES ALL PER UNIT LENGTH MATRICES GIVEN Z AND Y
MOPENF	MODIFIED OPENF PROCEDURE, PRIMARILY FOR RANDOM BINARY FILES
NORTCT	COMPUTES NORTON PARAMETER IN RECURSION FORM
OPDPUF*	FILE MANAGER FOR OPENING IASMAX DPULIJKN FILES (NOT IN LIBRARY)
OPNVOL	COMPUTES FULL N VECTOR VOLTAGE FOR N-M CURRENTS EQUAL ZERO
OPTADM	DETERMINES DPA AND VTR FOR ADMITTANCE WITH M OPEN TERMINALS
PROSEQ	DETERMINES NODE PROCESSING SEQUENCE, LEVELS AND CONSISTANCY
QCALIB	COMPUTES COEFFICIENTS FOR A SPLINED POLYNOMIAL FIT

RVEPRT	PRINTS OR WRITES FOR DIAGNOSIS A REAL VECTOR
THENCT	COMPUTES THENIVIN PARAMETER IN RECURSION FORM
TRANAD	COMPUTES DISTRISUTION TRANSFORMER ADMITTANCE FROM FILE TRANFILE
YADDEC	DECOMPOSES 3X3 OR 4X4 ANALYTIC MATPIX INTO DISCRETE PHYSICAL ELEMENTS
YADMIT	COMPUTES THE ANALYTIC ADMITTANCE ASSOCIATED WITH LUMPED ELEMENTS
YADMNU	CONVERTS NXN GROUNDED TO N+1XN+1 NON-PERFECT GROUNDED ADMITTANCE
YREALI	TESTS PHYSICAL REALIZABILITY OF A SPECIFIED ADMITTANCE MATRIX
YTRAMT	EXPANDS DISTRIBUTION TRANSFORMER SCALAR ADMITTANCE INTO MATRIX
YTRANP	COMPUTES DISTRIBUTION TRANSFORMER PRIMARY DRIVING POINT ADMITTANCE
ZPLOTM	SEMI GENERAL PURPOSE PLOTTING SUBROUTINE-M DEPENDANT VARIABLES
ZPLOTT	SEMI GENERAL PURPOSE PLOTTING SUBROUTINE-M DEPENDANT VARIABLES LONG X

```
**********************************************************************

LIBRGEN4                                          04/08/81    3:28 PM

**********************************************************************


0010$$N,ROUT(LB)
0020$:IDENT:R,RUSTAY1,RUSTAY
0030$:UTILITY
0040$:PRMFL:01,R,L,R,RUSTAY1/FORMFE01
0050$:PRMFL:02,R,L,R,RUSTAY1/AECDEN01
0060$:PRMFL:03,R,L,R,RUSTAY1/ADBMPY01
0070$:PRMFL:04,R,L,R,RUSTAY1/CDAPEC01
0080$:PRMFL:05,R,L,R,RUSTAY1/CHYPER01
0090$:PRMFL:06,R,L,R,RUSTAY1/CMAPEC01
0100$:PRMFL:07,R,L,R,RUSTAY1/CMTADD01
0110$:PRMFL:08,R,L,R,RUSTAY1/CMTCOP01
0120$:PRMFL:09,R,L,R,RUSTAY1/CMTCRC01
0130$:PRMFL:10,R,L,R,RUSTAY1/CMTIDE01
0140$:FILE:20,X1S,10L
0150$:FUTIL:01,20,MCOPY/1F/,HOLD/,20/
0160$:FUTIL:02,20,MCOPY/1F/,HOLD/,20/
0170$:FUTIL:03,20,MCOPY/1F/,HOLD/,20/
0180$:FUTIL:04,20,MCOPY/1F/,HOLD/,20/
0190$:FUTIL:05,20,MCOPY/1F/,HOLD/,20/
0200$:FUTIL:06,20,MCOPY/1F/,HOLD/,20/
0210$:FUTIL:07,20,MCOPY/1F/,HOLD/,20/
0220$:FUTIL:08,20,MCOPY/1F/,HOLD/,20/
0230$:FUTIL:09,20,MCOPY/1F/,HOLD/,20/
0240$:FUTIL:10,20,MCOPY/1F/
0250$:UTILITY
0260$:PRMFL:01,R,L,R,RUSTAY1/CMTIRC01
0270$:PRMFL:02,R,L,R,RUSTAY1/CMTMPY01
0280$:PRMFL:03,R,L,R,RUSTAY1/CMTPOL01
0290$:PRMFL:04,R,L,R,RUSTAY1/CMTPRT01
0300$:PRMFL:05,R,L,R,RUSTAY1/CMTRAN01
0310$:PRMFL:06,R,L,R,RUSTAY1/CMTSUB01
0320$:PRMFL:07,R,L,R,RUSTAY1/CMTTRP01
0330$:PRMFL:08,R,L,R,RUSTAY1/CSYADJ01
0340$:PRMFL:09,R,L,R,RUSTAY1/CVEPOL01
0350$:PRMFL:10,R,L,R,RUSTAY1/CVEPRT01
0360$:FILE:20,X2S,10L
0370$:FUTIL:01,20,MCOPY/1F/,HOLD/,20/
0380$:FUTIL:02,20,MCOPY/1F/,HOLD/,20/
0390$:FUTIL:03,20,MCOPY/1F/,HOLD/,20/
0400$:FUTIL:04,20,MCOPY/1F/,HOLD/,20/
0410$:FUTIL:05,20,MCOPY/1F/,HOLD/,20/
0420$:FUTIL:06,20,MCOPY/1F/,HOLD/,20/
0430$:FUTIL:07,20,MCOPY/1F/,HOLD/,20/
0440$:FUTIL:08,20,MCOPY/1F/,HOLD/,20/
0450$:FUTIL:09,20,MCOPY/1F/,HOLD/,20/
0460$:FUTIL:10,20,MCOPY/1F/
0470$:UTILITY
0480$:PRMFL:01,R,L,R,RUSTAY1/CZYRED01
0490$:PRMFL:02,R,L,R,RUSTAY1/ERRSTP01
```

```
0500$:PRMFL:03,R,L,R,RUSTAY1/LINEPU03
0510$:PRMFL:04,R,L,R,RUSTAY1/ZPLOTM01
0520$:PRMFL:05,R,L,R,RUSTAY1/CVDBAN01
0530$:PRMFL:06,R,L,R,RUSTAY1/PROSE002
0540$:PRMFL:07,R,L,R,RUSTAY1/SVEPRT01
0550$:PRMFL:08,R,L,R,RUSTAY1/THE:CT01
0560$:PRMFL:09,R,L,R,RUSTAY1/ERRSTT01
0570$:PRMFL:10,R,L,R,RUSTAY1/FILSTP01
0580$:FILE:20,X3S,10L
0590$:FUTIL:01,20,MCOPY/1F/,HOLD/,20/
0600$:FUTIL:02,20,MCOPY/1F/,HOLD/,20/
0610$:FUTIL:03,20,MCOPY/1F/,HOLD/,20/
0620$:FUTIL:04,20,MCOPY/1F/,HOLD/,20/
0630$:FUTIL:05,20,MCOPY/1F/,HOLD/,20/
0640$:FUTIL:06,20,MCOPY/1F/,HOLD/,20/
0650$:FUTIL:07,20,MCOPY/1F/,HOLD/,20/
0660$:FUTIL:08,20,MCOPY/1F/,HOLD/,20/
0670$:FUTIL:09,20,MCOPY/1F/,HOLD/,20/
0680$:FUTIL:10,20,MCOPY/1F/
0690$:UTILITY
0700$:PRMFL:01,R,L,R,RUSTAY1/YADMITO1
0710$:PRMFL:02,R,L,R,RUSTAY1/CMTDWD01
0720$:PRMFL:03,R,L,R,RUSTAY1/CPUFRW02
0730$:PRMFL:04,R,L,R,RUSTAY1/YTRANP02
0740$:PRMFL:05,R,L,R,RUSTAY1/CMTABC01
0750$:PRMFL:06,R,L,R,RUSTAY1/CCMMPY01
0760$:PRMFL:07,R,L,R,RUSTAY1/CMVMPY01
0770$:PRMFL:08,R,L,R,RUSTAY1/YREALI01
0780$:PRMFL:09,R,L,R,RUSTAY1/CMINRF01
0790$:PRMFL:10,R,L,R,RUSTAY1/YADMMU01
0800$:FILE:20,X4S,10L
0810$:FUTIL:01,20,MCOPY/1F/,HOLD/,20/
0820$:FUTIL:02,20,MCOPY/1F/,HOLD/,20/
0830$:FUTIL:03,20,MCOPY/1F/,HOLD/,20/
0840$:FUTIL:04,20,MCOPY/1F/,HOLD/,20/
0850$:FUTIL:05,20,MCOPY/1F/,HOLD/,20/
0860$:FUTIL:06,20,MCOPY/1F/,HOLD/,20/
0870$:FUTIL:07,20,MCOPY/1F/,HOLD/,20/
0880$:FUTIL:08,20,MCOPY/1F/,HOLD/,20/
0890$:FUTIL:09,20,MCOPY/1F/,HOLD/,20/
0900$:FUTIL:10,20,MCOPY/1F/
0910$:UTILITY
0920$:PRMFL:01,R,L,R,RUSTAY1/CMTCPY01
0930$:PRMFL:02,R,L,R,RUSTAY1/CMTZR001
0940$:PRMFL:03,R,L,R,RUSTAY1/CMTMPT01
0950$:PRMFL:04,R,L,R,RUSTAY1/YTRAMT02
0960$:PRMFL:05,R,L,R,RUSTAY1/CMAN3C01
0970$:PRMFL:06,R,L,R,RUSTAY1/TRANAD04
0980$:PRMFL:07,R,L,R,RUSTAY1/YADDEC01
0990$:PRMFL:08,R,L,R,RUSTAY1/OPNVOL01
1000$:PRMFL:09,R,L,R,RUSTAY1/CMDBAN01
1010$:PRMFL:10,R,L,R,RUSTAY1/MORTCT01
```

```
1020$:FILE:20,X5S,10L
1030$:FUTIL:01,20,*COPY/1F/,HOLD/,20/
1040$:FUTIL:02,20,*COPY/1F/,HOLD/,20/
1050$:FUTIL:03,20,*COPY/1F/,HOLD/,20/
1060$:FUTIL:04,20,*COPY/1F/,HOLD/,20/
1070$:FUTIL:05,20,*COPY/1F/,HOLD/,20/
1080$:FUTIL:06,20,*COPY/1F/,HOLD/,20/
1090$:FUTIL:07,20,*COPY/1F/,HOLD/,20/
1100$:FUTIL:08,20,*COPY/1F/,HOLD/,20/
1110$:FUTIL:09,20,*COPY/1F/,HOLD/,20/
1120$:FUTIL:10,20,*COPY/1F/
1130$:UTILITY
1140$:PRMFL:01,R,L,5,RUSTAY1/CMTCMP01
1150$:PRMFL:02,R,L,5,RUSTAY1/CMECMP01
1160$:PRMFL:03,R,L,5,RUSTAY1/MOPENF01
1170$:PRMFL:04,R,L,5,RUSTAY1/ZPLOTT01
1171$:PRMFL:05,R,L,5,RUSTAY1/OPTADM01
1180$:FILE:20,X6S,10L
1190$:FUTIL:01,20,*COPY/1F/,HOLD/,20/
1200$:FUTIL:02,20,*COPY/1F/,HOLD/,20/
1210$:FUTIL:03,20,*COPY/1F/,HOLD/,20/
1211$:FUTIL:04,20,*COPY/1F/,HOLD/,20/
1220$:FUTIL:05,20,*COPY/1F/
1230$:UTILITY
1240$:FILE:01,X1R,10L
1250$:FILE:02,X2R,10L
1260$:FILE:03,X3R,10L
1270$:FILE:04,X4R,10L
1280$:FILE:05,X5R,10L
1290$:FILE:06,X6R,10L
1300$:FILE:07,X7S,10L
1310$:FUTIL:01,07,RWD/01,07/,*COPY/1F/,HOLD/,07/
1320$:FUTIL:02,07,RWD/02,/,*COPY/1F/,HOLD/,07/
1330$:FUTIL:03,07,RWD/03,/,*COPY/1F/,HOLD/,07/
1340$:FUTIL:04,07,RWD/04,/,*COPY/1F/,HOLD/,07/
1350$:FUTIL:05,07,RWD/05,/,*COPY/1F/,HOLD/,07/
1360$:FUTIL:06,07,RWD/06,/,COPY/1F/
1370$:PROGRAM:RAMLIB
1380$:FILE:R*,X7R,10L
1390$:FILE:A4,X2D,7R,NEW,LIBRARY4
1400$:ENDJOB
```

```
ABCDENS1                                                    04/08/81     3:52 PM

010**RUN *=:ABCDEN01(NOGO,CORE=30K)
020        SUBROUTINE ABCDEN(DL,ZO,YO,S,SI,X,N,
030       &                        AM,BM,AO,BO,CO,DO,SA1)
040 *        THIS SUBROUTINE COMPUTES THE AO,BO,CO,DO
050 *        COEFFFICIENT MATRICES FOR A UNIFORM N CONDUCTOR
060 *        TRANSIMISSION LINE. IT INPUTS THE EIGENVALUE VECTOR
070 *        DL, THE CHARACTERISTIC IMPEDANCE ZO AND ADMITTANCE
080 *        YO MATRICES (IN PHASE CONDUCTOR CORDINATES) AND THE
090 *        VOLTAGE MODAL MATRIX S AND ITS INVERSE SI
100        COMPLEX DL(N),AM(N),BM(N),ZERO,ONE
110        COMPLEX ZO(N,N),AO(N,N),CO(N,N),S(N,N),SI(N,N)
120        COMPLEX YO(N,N),BO(N,N),DO(N,N),SA1(N,N)
130        DATA ZERO/(0.,0.)/,ONE/(1.,0.)/
140        IF(N.GT.M)CALL ERRSTP('ABCDEN',0)
150        IF(X.NE.(0.))GOTO200
160 *        FOR X=0, A=D=I, C=D=NULL
170        DO 100 I=1,N
180        DO 100 J=1,N
190        AO(I,J)=ZERO
200        BO(I,J)=ZERO
210        CO(I,J)=ZERO
220        DO(I,J)=ZERO
230        IF(I.NE.J)GOTO100
240        AO(I,J)=ONE
250        DO(I,J)=ONE
260    100 CONTINUE
270        RETURN
280    200 CONTINUE
290        DO 300 I=1,N
300    300 CALL CHYPER(DL(I)*X,BM(I),AM(I))
310        CALL ADBMPY(S,AM,SI,AO,N,M)
320        CALL CMTRAN(AO,DO,N,N,M)
330        CALL ADBMPY(S,BM,SI,SA1,N,M)
340        CALL CMTMPY(SA1,ZO,BO,N,N,N,M)
350        CALL CMTMPY(YO,SA1,CO,N,N,M,M)
360        RETURN
370        END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
010**RUN *=;ADB:PYS1(NOGO,CORE=30K)
020       SUBROUTINE ADBMPY(A,D,B,C,N,M)
030 *       THIS SUBROUTINE COMPUTES THE SPECIAL COMPLEX MATRIX PRODUCT
040 *       C=A*D*B WHERE A,B,C USE THE UPPER LEFT NXN PARTITIONS OF
050 *       RESPECTIVE MXM MATRICES. D IS A COMPLEX VECTOR OF DIMENSION
060 *       M OF WHICH THE UPPER N ELEMENTS REPPESENT A DIAGONAL MATRIX
070       COMPLEX A(M,M),B(M,M),C(M,M),D(M),SUM,ZERO
080       DATA ZERO/(0.,0.)/
090       IF(N.GT.M)CALL ERRSTP('ADBMPY',1)
100       DO 200 I=1,N
110       DO 200 J=1,N
120       SUM=ZERO
130       DO 100 K=1,N
140   100 SUM=SUM+A(I,K)*D(K)*B(K,J)
150   200 C(I,J)=SUM
160       RETURN
170       END
```

```
010**RUN  *=;CCMMPYS1(NOGO,CORE=30K)
020       SUBROUTINE CCMMPY(A1,B1,C1,D1,A2,B2,C2,D2,A3,B3,C3,D3,SA,N,M,IDIA,LU
030 *       THIS SUBROUTINE PERFORMS THE COMPLEX CHAIN MATRIX MULTIPLY
040 *         A3=A1*A2+B1*C2
050 *         B3=A1*B2+B1*D2
060 *         C3=C1*A2+D1*C2
070 *         D3=C1*B2+D1*D2
080 *       WHERE EACH MATRIX IS SQUARE NXN AND EACH DIMENSIONED MXM
090 *       SA IS A SCRATCH ARRAY
100     10 FORMAT('IDENTITY CHECK IN CCMMPY')
110       COMPLEX A1(M,M),B1(M,M),C1(M,M),D1(M,M)
120       COMPLEX A2(M,M),B2(M,M),C2(M,M),D2(M,M)
130       COMPLEX A3(M,M),B3(M,M),C3(M,M),D3(M,M)
140       COMPLEX SA(M,M),SUM,ZERO,ONE
150       CHARACTER SUB*6
160       DATA ZERO/(0.,0.)/,ONE/(1.,0.)/,EPSI/1.E-6/,SUB/'CCMMPY'/
170       IF(N.GT.M) CALL ERRSTP(SUB,0)
180       CALL CMTMPY(A1,A2,A3,N,M,N,M)
190       CALL CMTMPY(B1,C2,SA,N,M,N,M)
200       CALL CMTADD(A3,SA,A3,N,M)
210       CALL CMTMPY(A1,B2,B3,N,M,N,M)
220       CALL CMTMPY(B1,D2,SA,N,M,N,M)
230       CALL CMTADD(B3,SA,B3,N,M)
240       CALL CMTMPY(C1,A2,C3,N,M,N,M)
250       CALL CMTMPY(D1,C2,SA,N,M,N,M)
260       CALL CMTADD(C3,SA,C3,N,M)
270       CALL CMTMPY(C1,B2,D3,N,M,N,M)
280       CALL CMTMPY(D1,D2,SA,N,M,N,M)
290       CALL CMTADD(D3,SA,D3,N,M)
300       IF(IDIA.EQ.0)RETURN
310 *       IF DESIRED CHECK IDENITY AT*D-CT*B=I
320 *       FIRST FORM SA=AT*D-CT*B
330       DO 200 I=1,N
340       DO 200 J=1,N
350       SUM=ZERO
360       DO 100 K=1,N
370   100 SUM=SUM+A3(K,I)*D3(K,I)-C3(K,I)*B3(K,I)
380   200 SA(I,J)=SUM
390 *       SA SHOULD BE DIAGONAL.  COULD CALL CMTIDE.  WILL OPT FOR
400 *       LOCAL CHECK FOR COMMUNICATION CONVENIENCE
410 *       FIRST CHECK DIAGONALS
420       DO 300 I=1,N
430   300 IF(CABS(SA(I,I)).GT.EPSI)GOTO600
440 *       NEXT CHECK OFF DIAGONAL ELEMENTS
450       XMAX=0.
460       DO 400 I=1,N-1
470       DO 400 J=I+1,N
480       X=AMAX1(CABS(SA(I,J)),CABS(SA(J,I)))
490       IF(X.GT.XMAX)XMAX=X
```

```
500    400 CONTINUE
510        IF(X.LT.EPSI)RETURN
520    600 CONTINUE
530        WRITE(LU,10)
540        CALL CMTPRT(SA,N,M,LU)
550        CALL ERRSTP(SUB,1)
560        STOP
570        END
```

CDAPBCS1 04/08/81 3:52 PM

```
010*#RUN +=:CDAPSC01(NOGO,CORE=30K)
020       SUBROUTINE CDAPBC(AD,BD,C,W,N,M)
030 *       THIS SUBROUTINE COMPUTES THE SPECIAL MATRIX OPERATION
040 *       W=AD+BD*C WHERE AD,BD ARE DIAGONAL AND REPRESENTED BY THE
050 *       UPPER 1 ELEMENTS OF N-VECTORS.  W AND C ARE THE UPPER LEFT
060 *       NXN PARTITION OF GENERAL MXM MATRICES.  SEE RCR E-8A,
070 *       E-16A,E-17A
080       COMPLEX W(M,M),C(M,M),AD(M),BD(M),YI,YK
090 *       CALCULATE DIAGONAL TERMS (WHICH INVOLVE AD)
100       IF(N.GT.M)CALL ERRSTP('CDAPBC',1)
110       DO 100 I=1,N
120   100 W(I,I)=AD(I)+BD(I)*C(I,I)
130 *       NEXT CALCULATE OFF DIAGONAL TERMS (WHICH DO NOT INVOLVE AD)
140       NP1=N+1
150       DO 200 I=1,N-1
160       K=NP1-I
170       YI=BD(I)
180       YK=BD(K)
190       DO 200 J=I+1,N
200       L=NP1-J
210       W(I,J)=YI*C(I,J)
220   200 W(K,L)=YK*C(K,L)
230       RETURN
240       END
```

```
CHYPERS1                                                    04/08/81     3:53 PM

010**RUN *=:CHYPER01(NOGO,CORE=30K)
020        SUBROUTINE CHYPER(Z,SINH,COSH)
030 *        THIS SUBROUTINE COMPUTES THE COMPLEX HYPERBOLIC SINE AND
040 *        COSINE OF COMPLEX ARGUMENT Z.  THIS COMPUTATIONAL ARRANGEMENT
050 *        WAS CHOSEN FOR NUMERICAL CONSIDERATIONS
060        COMPLEX Z,SINH,COSH
070        DATA HALF/.5/,ONE/1./
080        X= REAL(Z)
090        Y=AIMAG(Z)
100        U=EXP(X)
110        RCOSH=HALF*(U+ONE/U)
120        RSINH=RCOSH*TANH(X)
130        RSIN=SIN(Y)
140        RCOS=COS(Y)
150        SINH=CMPLX(RSINH*RCOS,RCOSH*RSIN)
160        COSH=CMPLX(RCOSH*RCOS,RSINH*RSIN)
170        RETURN
180        END
```

```
**********************************************************************

CMAMBCS1                                          04/08/81     3:53 PM

**********************************************************************


010**RUN*=;CMAMBCS1(..OGO,CORE=30K)
020       SUBROUTINE CMAMBC(A,B,C,D,L,M,N,NDIM)
030 *       THIS SUBROUTINE COMPUTES THE SPECIAL COMPLEX MATRIX OPERATION
040 *       D=A-B*C WHERE EACH MATRIX IS DIMENSIONED NDIMXNDIM
050 *       AND WHERE
060 *         A IS LXN
070 *         B IS LXM
080 *         C IS MXN
090 *         D IS LXN
100       COMPLEX A(NDIM,NDIM),B(NDIM,NDIM),C(NDIM,NDIM),D(NDIM,NDIM)
110       COMPLEX SUM
120       IF(MAX0(L,M,N).GT.NDIM)CALL ERRSTT('CMAMBC',1,L,M,N)
130       DO 200 I=1,L
140       DO 200 J=1,N
150       SUM=A(I,J)
160       DO 100 K=1,M
170   100 SUM=SUM-B(I,K)*C(K,J)
180   200 D(I,J)=SUM
190       RETURN
200       END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
CMAPBCS1                                                    04/08/81     3:53 PM

010**RUN*=;CMAPBCS1(NOGO,CORE=30K)
020        SUBROUTINE CMAPBC(A,B,C,D,L,M,N,NDIM)
030 *        THIS SUBROUTINE COMPUTES THE SPECIAL COMPLEX MATRIX OPERATION
040 *        D=A*B*C WHERE EACH MATRIX IS DIMENSIONED NDIMXNDIM
050 *        AND WHERE
060 *          A IS LXM
070 *          B IS LXN
080 *          C IS NXM
090 *          D IS LXM
100        COMPLEX A(NDIM,NDIM),B(NDIM,NDIM),C(NDIM,NDIM),D(NDIM,NDIM)
110        COMPLEX SUM
120        IF(MAXO(L,M,N).GT.NDIM)CALL ERRSTT('CMAPBC',1,L,M,N)
130        DO 200 I=1,L
140        DO 200 J=1,M
150        SUM=A(I,J)
160        DO 100 K=1,N
170    100 SUM=SUM+B(I,K)*C(K,J)
180    200 D(I,J)=SUM
190        RETURN
200        END
```

```
**********************************************************************

CMDBANS1                                          04/08/81     3:53 PM

**********************************************************************

010**RUN *=:CMDBAN.01(NOGO,CORE=30K)
020        SUBROUTINE CMDBAN(A,N,M,MN,LU,WKAREA)
030  *       THIS SUBROUTINE PRINTS OR WRITES FOR DIAGNOSTIC PURPOSES
040  *       THE ELEMENTS OF A SQUARE COMPLEX MATRIX IN POLAR DB FORM
050  *         LU.EQ.06 FOR PRINT
060  *         LU.NE.06 FOR WRITE TO FILE
070       11 FORMAT(9F9.2)
080       12 FORMAT(' ')
090          COMPLEX A(M,M),Z
100          DIMENSION WKAREA(MX)
110          DATA DEG/57.29578/,RZERO/0./
120          IF(N.GT.M.OR.(N+N).GT.MX)CALL ERRSTP('CMTPOL',1)
130          DO 200 I=1,N
140          DO 190 J=1,N
150          Z=A(I,J)
160          AMP=CABS(Z)
170          IF(AMP.LE.RZERO)GOTO150
180          DB=20.*ALOG10(AMP)
190          WKAREA(J)=DB
200          WKAREA(J+N)=DEG*ATAN2(AIMAG(Z),REAL(Z))
210          GOTO190
220       150 WKAREA(J)=-9999.99
230          WKAREA(J+N)=RZERO
240       190 CONTINUE
250          WRITE(LU,11)(WKAREA(J),J=1,N)
260          WRITE(LU,11)(WKAREA(J+N),J=1,N)
270       200 WRITE(LU,12)
280          RETURN
290          END
```

```
010**RUN *=:CMI.RFS1('LOGO,CORE=30K)
020       SUBROUTINE CMINRF(A,SA1,SA2,SA3,N,M,WKAREA,NRUN)
030 *       THIS SUBROUTINE FIRST COMPUTES AN ESTIMATE OF
040 *       AN INVERSE OF MATRIX A USING LIBRARY ROUTINE CMTINV.
050 *       IF NRUN.GT.0 THEN NRUN ITERATIONS USING FADDEEVA'S METHOD
060 *       ARE USED TO IMPROVE THE ESTIMATE.  DOUBLE BUFFERING
070 *       TECHNIQUE IS USED TO SAVE COPY PROCEDURE.
080       COMPLEX A(N,M),SA1(N,M),SA2(N,M),SA3(N,M),TWO
090       DATA TWO/(2.,0.)/
100       IFLIP=0
110       CALL CMTCOP(A,SA1,N,M)
120       CALL CMTINV(SA1,N,N,M,WKAREA)
130       IF(NRUN.EQ.0)GOTO250
140       DO 200 NRUN=1,NRUN
150       IF(IFLIP.EQ.1)GOTO150
160 *       AT THIS POINT SA1 CONTAINS CURRENT ESTIMATE OF AINVERSE
170       CALL CMTMPY(A,SA1,SA2,N,M,N,M)
180       DO 100 I=1,N
190       SA2(I,I)=TWO-SA2(I,I)
200       DO 100 J=1,M
210   100 IF(I.NE.J)SA2(I,J)=-SA2(I,J)
220       CALL CMTMPY(SA1,SA2,SA3,N,M,N,M)
230       IFLIP=1
240 *       AT THIS POINT SA3 CONTAINS CURRENT ESTIMATE OF AINVERSE
250       GOTO200
260   150 CONTINUE
270 *       AT THIS POINT SA3 CONTAINS CURRENT ESTIMATE OF AINVERSE
280       CALL CMTMPY(A,SA3,SA2,N,M,N,M)
290       DO 175 I=1,N
300       SA2(I,I)=TWO-SA2(I,I)
310       DO 175 J=1,M
320   175 IF(I.NE.J)SA2(I,J)=-SA2(I,J)
330       CALL CMTMPY(SA3,SA2,SA1,N,M,N,M)
340       IFLIP=0
350 *       AT THIS POINT SA1 CONTAINS THE CURRENT ESTIMATE OF AINVERSE
360   200 CONTINUE
370       PRINT,'COMPLETED ',NRUN,' ITERATIONS'
380   250 IF(IFLIP.EQ.0)CALL CMTCOP(SA1,A,N,M)
390       IF(IFLIP.EQ.1)CALL CMTCOP(SA3,A,N,M)
400   300 RETURN
410       END
```

```
CMTABCS1                                                    04/08/81     3:54 PM

010**RUN **:CMTABCS1(ACCO,CORE=30K)
020       SUBROUTINE CMTABC(A,B,C,L,M,N,NDIM)
030 *       THIS SUBROUTINE PERFORMS THE COMPLEX TRANSPOSE
040 *       BCONJUGATE MATRIX MULTIPLY C=ATRANSPOSE*BCONJUGATE WITH
050 *       EACH DIMENSIONED NDIM*NDIM AND WHERE
060 *         A IS LXL
070 *         B IS LXN
080 *         C IS LXN
090       COMPLEX A(NDIM,NDIM),B(NDIM,NDIM),C(NDIM,NDIM),SUM,ZERO,TEMP
100       DATA ZERO/(0.,0.)/
110       IF(MAX0(L,M,N).GT.NDIM)CALL ERRSTP('CMTABC',1)
120       DO 200 I=1,L
130       DO 200 J=1,N
140       SUM=ZERO
150       DO 100 K=1,M
160       TEMP=B(K,J)
170   100 SUM=SUM+A(K,I)*CMPLX(REAL(TEMP),-AIMAG(TEMP))
180   200 C(I,J)=SUM
190       RETURN
200       END
```

```
**********************************************************************

CMTADDS1                                          04/08/81     3:54 PM

**********************************************************************


010**RUN *$CMTADDS1(NOGO,CORE=30K)
020        SUBROUTINE CMTADD(A,B,C,N,M)
030 *        THIS SUBROUTINE PERFORMS THE COMPLEX MATRIX ADD
040 *        C=A+B USING THE UPPER LEFT NXN PARTITION OF MATRICES
050 *        A,B,C WHICH ARE EACH DIMENSIOD MXM
060 *        NOTE THIS ROUTINE CAN BE DONE IN PLACE, IE A=A+B
070        COMPLEX A(M,M),B(M,M),C(M,M)
080        IF(N.GT.M)CALL ERRSTP('CMTADD',1)
090        DO 100 I=1,N
100        DO 100 J=1,N
110    100 C(I,J)=A(I,J)+B(I,J)
120        RETURN
130        END
```

```
CMTCMPS1                                                    04/08/81     3:54 PM

010**RUN *=$CMTCMPS1(,LOGO,CORE=30K)
020       SUBROUTINE CMTCMP(A,R,SA1,N,NDIM,ID)
030 *       THIS SUBROUTINE COMPARES AN INPUT COMPLEX SQUARE MATRIX A
040 *       WITH A REFERENCE COMPLEX MATRIX R WHOSE ELEMENT NUMERICAL
050 *       VALUES ARE ESTABLISHED BY THE INITIAL EXECUTION OF THIS
060 *       SUBROUTINE
070     10 FORMAT(' ')
080     11 FORMAT(8F15.6)
090     12 FORMAT(A43,F15.6)
100        COMPLEX ZERO,TEMP
110        COMPLEX A(NDIM,NDIM),R(NDIM,NDIM),SA1(NDIM,NDIM)
120        DATA IFLAG/0/,RZERO/0./,ZERO/(0.,0.)/
130        IF(IFLAG.EQ.1)GOTO200
140        CALL CMTCOP(A,R,N,NDIM)
150        IFLAG=1
160        IF(N.EQ.NDIM)GOTO200
170        CALL CMTZRO(R,N,N,NDIM)
180   200  CONTINUE
190        XII=RZERO
200        XIJ=RZERO
210        DO 220 I=1,N
220        DO 220 J=1,N
230        X=CABS(R(I,J))
240        IF(X.EQ.RZERO)GOTO210
250        TEMP=(A(I,J)-R(I,J))/X
260        SA1(I,J)=TEMP
270        X=CABS(TEMP)
280        IF(I.EQ.J.AND.X.GT.XII)XII=X
290        IF(I.NE.J.AND.X.GT.XIJ)XIJ=X
300        GOTO220
310   210  SA1(I,J)=ZERO
320   220  CONTINUE
330        CALL FORMFE
340        PRINT,'ID=',ID
350        PRINT,'PERTURBED MATRIX'
360        CALL CMTPRT(A,N,NDIM,06)
370        PRINT 10
380        PRINT,'REFERENCE MATRIX'
390        CALL CMTPRT(R,N,NDIM,06)
400        PRINT 10
410        PRINT,'NORMALIZED DIFFERENCE MATRIX'
420        PRINT 10
430        DO 300 I=1,N
440        PRINT 11,( REAL(SA1(I,J)),J=1,N)
450   300  PRINT 11,(AIMAG(SA1(I,J)),J=1,N)
460        PRINT 10
470        PRINT 12,'MAXIMUM NORMALIZED  ON DIAGONAL DIFFERENCE=',XII
480        PRINT 12,'MAXIMUM NORMALIZED OFF DIAGONAL DIFFERENCE=',XIJ
490        CALL FORMFE
```

```
500         RETURN
510         END
```

CMTCOPS1 04/08/81 3:54 PM

```
010**RUN *=;CMTCOP01(NOGO,CORE=30K)
020       SUBROUTINE CMTCOP(A,B,N,M)
030 *       THIS SUBROUTINE PERFORMS THE COPY FUNCTION B=A FOR THE
040 *       UPPER LEFT NXN PARTITION OF MXM A AND B.
050       COMPLEX A(M,M),B(N,M)
060       IF(N.GT.M)CALL ERRSTP('CMTCOP',1)
070       DO 100 I=1,N
080       DO 100 J=1,N
090   100 B(I,J)=A(I,J)
100       RETURN
110       END
```

```
010**RUN *=;CMTCPY01(*,GO,CORE=30K)
020       SUBROUTINE CMTCPY(A,B,M,N,NDIM)
030 *       THIS SUBROUTINE COPIES A INTO B IE B=A AND ALSO
040 *       ZEROES THE UNUSED BORDER ELEMENTS OF B
050 *       EACH MATRIX IS DIMENSIONED NDIMXNDIM.
060       COMPLEX A(NDIM,NDIM),B(NDIM,NDIM)
070       IF(M.GT.NDIM.OR.N.GT.NDIM)CALL ERRSTP('CMTCPY',1)
080       DO 200 I=1,NDIM
090       DO 200 J=1,NDIM
100       IF(I.GT.M.OR.J.GT.N)GOTO100
110       B(I,J)=A(I,J)
120       GOTO200
130   100 B(I,J)=(0.,0.)
140   200 CONTINUE
150       RETURN
160       END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
010**RUN *=:CMTCRCS1(*IOGO,CORE=30K)
020       SUBROUTINE CMTCRC(A,B,K,L,N,M)
030 *       THIS SUBROUTINE COPIES THE UPPER LEFT PARTITION OF COMPLEX
040 *       MATRIX A TO B WITH THE K AND L ROWS AND COLUMNS INTERCHANGED
050 *       NOTE K.LE.N.AND.L.LE.N.AND.N.LE.M
060 *       NOTE CALLS SUBROUTINES CMTCOP AND CMTIRC
070       COMPLEX A(*,M),B(*,M)
080       IF(K.GT.N.OR.L.GT.N.OR.N.GT.M)CALL ERRSTP('CMTCRC',1)
090       CALL CMTCOP(A,B,M,M)
100       CALL CMTIRC(B,K,L,N,M)
110       RETURN
120       END
```

```
CMTDWDS1                                                    04/08/81     3:54 PM

010**RUN **,CMTDWDS1(NOGO,CORE=30K)
020        SUBROUTINE CMTDWD(AD,BW,CD,DW,N,M)
030 *        THIS SUBROUTINE COMPUTES THE SPECIAL COMPLEX MATRIX
040 *        PRODUCT DW=AD*BW*CD WHERE AD AND CD ARE DIAGONAL
050 *        MATRICES REPRESENTED IN N VECTOR PARTITION OF AN
060 *        N VECTOR AND DW AND BW ARE THE N*N UPPER LFT
070 *        PARTITIONS OF N*N MATRICES.  THE N-M ROWS AND COLUMNS
080 *        OF THE OUTPUT MATRIX DW ARE ZERO FILLED.
090        COMPLEX AD(1),BW(N,1),CD(1),DW(N,N),ZERO
100        DATA ZERO/(0.,0.)/
110        IF(N.GT.M)CALL ERRSTP('CMTDWD',1)
120        DO 100 I=1,N
130        DO 100 J=1,M
140        IF((I.LE.M).AND.(J.LE.M))DW(I,J)=AD(I)*BW(I,J)*CD(J)
150        IF((I.GT.M).OR.(J.GT.M))DW(I,J)=ZERO
160    100 CONTINUE
170        RETURN
180        END
```

```
**********************************************************************

CMTIDES1                                          04/08/81    3:55 PM.

**********************************************************************


010**RUN *=:CMTIDE01(NOGO,CORE=35K)
020       SUBROUTINE CMTIDE(A,N,M,IIDE,IDIA,LU)
030 *       THIS SUBROUTINE PERFORMS A CHECK TO SEE IF THE UPPER
040 *       LEFT NXN PARTITION OF THE MXM COMPLEX MATRIX A IS
050 *       DIAGONAL OR IDENTITY. THE CONTROL PARAMETERS ARE
060 *         IIDE.EQ.0 PERFORM DIAGONAL CHECK
070 *         IIDE.NE.0 PERFORM IDENTITY CHECK
080 *         LU.EQ.06 FOR DIAGNOSTIC PRINT
090 *         LU.NE.06 FOR DIAGNOSTIC WRITE TO FILE
100 *       IF TESTS ARE NEGATIVE THEN SUBROUTINE CMTPRT IS CALLED
110 *       AND STOPS BY CALL TO ERRSTP
120     10 FORMAT('XMAX IN CMTIDE=',1PE14.7)
130       COMPLEX A(M,M),ONE,Z
140       DATA EPSI/1.E-6/,ONE/(1.,0.)/,RZERO/0./,RONE/1./
150       DATA EPST/1.E-4/,XLAR/1.E6/
160       IF(N.GT.M)CALL ERRSTP('CMTIDE',1)
170       XMIN=RONE
180       IF(IIDE.EQ.0)GOTO200
190 *       CHECK DIAGONAL ELEMENTS FOR UNITY FOR IDENTITY CHECK
200       DO 100 I=1,N
210    100 IF(CABS(A(I,I)-ONE).GT.EPSI)GOTO600
220       GOTO400
230 *       IN ORDER TO SAVE CORE, FIND SMALLEST DIAGONAL ELEMENT MAGNITUDE
240    200 XMIN=XLAR
250       DO 300 I=1,N
260       X=CABS(A(I,I))
270    300 IF(X.LT.XMIN)XMIN=X
280 *       NOW COMPARE OFF DIAGONAL ELEMENTS FOR EITHER CASE
290 *         XMIN=RONE FOR IDENTITY CHECK
300 *         XMIN=SMALLEST DIAGONAL ELEMENT MAGNITUDE FOR DIAGONAL CHECK
310    400 CONTINUE
320       XMAX=RZERO
330       DO 500 I=1,N-1
340       DO 500 J=I+1,N
350       X=AMAX1(CABS(A(I,J)),CABS(A(J,I)))/XMIN
360       IF(X.GT.XMAX)XMAX=X
370    500 CONTINUE
380       IF(IDIA.NE.0)WRITE(LU,10)XMAX
390       IF(XMAX.GT.EPST)GOTO600
400       RETURN
410    600 CALL CMTPRT(A,N,M,LU)
420       PRINT,'XMAX IN CMTIDE=',XMAX
430       RETURN
440       END
```

```
CMTIRCS1                                                    04/08/81     3:55 PM

010**RUN *=;CMTIRC01(NOGO,CORE=30K)
020        SUBROUTINE CMTIRC(A,K,L,N,M)
030 *        THIS SUBROUTINE INTERCHANGES THE K AND L COLUMNS AND ROWS OF
040 *        COMPLEX MATRIX A WITH L.LE.N.AND.K.LE.N.
050 *        NOTE THE OPERATION IS DONE IN PLACE, IE A TO A
060        COMPLEX A(M,M),C
070        IF(N.GT.M.OR.L.GT.N.OR.K.GT.N)CALL ERRSTP('CMTINT',1)
080 *        INTERCHANGE ROWS
090        DO 100 J=1,N
100        C=A(K,J)
110        A(K,J)=A(L,J)
120    100 A(L,J)=C
130 *        INTERCHANGE COLUMNS
140        DO 200 I=1,N
150        C=A(I,K)
160        A(I,K)=A(I,L)
170    200 A(I,L)=C
180        RETURN
190        END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
**********************************************************************

CMTMPTS1                                          04/08/81     3:55 PM

**********************************************************************


010*#RUN *=;CMTMPTD1(LOGO,CORE=30K)
020       SUBROUTINE CMTMPT(A,B,C,L,M,N,NDIM)
030 *       THIS SUBROUTINE PERFORMS THE COMPLEX MATRIX TRANSPOSE
040 *       MULTIPLY C=A*BT WITH EACH DIMENSIONED NDIMXNDIM AND WHERE
050 *         A IS MXL
060 *         B IS NXL
070 *         C IS MXN
080      *  COMPLEX A(NDIM,NDIM),B(NDIM,NDIM),C(NDIM,NDIM),SUM,ZERO
090       DATA ZERO/(0.,0.)/
100       IF(L.GT.NDIM)CALL ERRSTP('CMTMPT',1)
110       DO 200 I=1,M
120       DO 200 J=1,N
130       SUM=ZERO
140       DO 100 K=1,L
150  100  SUM=SUM+A(I,K)*B(J,K)
160  200  C(I,J)=SUM
170       RETURN
180       END
```

```
**********************************************************************

CMTMPYS1                                              04/08/81     3:55 PM

**********************************************************************


010**RUN *S;CMTMPYO1(LOGO,CORE=30K)
020       SUBROUTINE CMTMPY(A,B,C,L,M,N,NDIM)
030 *       THIS SUBROUTINE PERFORMS THE COMPLEX MATRIX MULTIPLY
040 *       C=A*B WITH EACH DIMENSIONED NDIMXNDIM AND WHERE
050 *         A IS LXM
060 *         B IS MXN
070 *         C IS LXN
080       COMPLEX A(NDIM,NDIM),B(NDIM,NDIM),C(NDIM,NDIM),SUM,ZERO
090       DATA ZERO/(0.,0.)/
100       IF(MAXO(L,M,N).GT.NDIM)CALL ERRSTP('CMTMPY',1)
110       DO 200 I=1,L
120       DO 200 J=1,N
130       SUM=ZERO
140       DO 100 K=1,M
150   100 SUM=SUM+A(I,K)*B(K,J)
160   200 C(I,J)=SUM
170       RETURN
180       END
```

```
CMTPOLS1                                                    04/08/81    3:56 PM

010**RUN *=:CMTPOLS1(:,OGO,CORE=30K)
020       SUBROUTINE CMTPOL(A,N,M,NW,LU,WKAREA)
030 *       THIS SUBROUTINE PRINTS OR WRITES FOR DIAGNOSTIC PURPOSES
040 *       THE ELEMENTS OF A SQUARE COMPLEX MATRIX IN POLAR FORM
050 *         LU.EQ.06 FOR PRINT
060 *         LU.NE.06 FOR WRITE TO FILE
070 *       NOTE IF N.GT.8 PROBABLY BEST TO REDUCE 15 FIELD WIDTH
080     11 FORMAT(1PE15.7)
090     12 FORMAT(' ')
100     13 FORMAT(9F15.7)
110        COMPLEX A(M,M),Z
120        DIMENSION WKAREA(1,W)
130        DATA DEG/57.29578/
140        IF(N.GT.M.OR.(N+N).GT.NW)CALL ERRSTP('CMTPOL',1)
150        DO 200 I=1,N
160        DO 100 J=1,N
170        Z=A(I,J)
180        X= REAL(Z)
190        Y=AIMAG(Z)
200        WKAREA(J)=CABS(Z)
210    100 WKAREA(J+N)=DEG*ATAN2(Y,X)
220        WRITE(LU,11)(WKAREA(J),J=1,N)
230        WRITE(LU,13)(WKAREA(J+N),J=1,N)
240    200 WRITE(LU,12)
250        RETURN
260        END
```

CMTPRTS1 04/08/81 3:56 PM

```
010**RUN *=:CMTPRTO1(:.OGO,CORE=30K)
020        SUBROUTINE CMTPRT(A,N,M,LU)
030 *        THIS SUBROUTINE PRINTS OR WRITES FOR DIAGNOSTIC PURPOSES
040 *        THE ELEMENTS OF A SQUARE COMPLEX MATRIX IN RECTANGULAR FORM
050 *          LU.EQ.06 FOR PRINT
060 *          LU.NE.06 FOR WRITE TO FILE
070 *        NOTE IF N.GT.6 PROBABLY BEST TO REDUCE 15 FIELD WIDTH
080     11 FORMAT(1P6E15.7)
090     12 FORMAT(' ')
100        COMPLEX A(M,N)
110        IF(N.GT.M)CALL ERRSTP('CMTPRT',1)
120        DO 100 I=1,N
130        WRITE(LU,11)( REAL(A(I,J)),J=1,N)
140        WRITE(LU,11)(AIMAG(A(I,J)),J=1,N)
150    100 WRITE(LU,12)
160        RETURN
170        END
```

```
CMTRANS1                                                    04/08/81     3:56 PM
```

```
010**RUN **$CMTRAN01(NOGO,CORE=30K)
020        SUBROUTINE CMTRAN(A,B,M,N,NDIM)
030  *       THIS SUBROUTINE COMPUTES THE TRANSPOSE OF A COMPLEX MATRIX
040  *       B=AT WITH EACH DIMENSIONED NDIMXNDIM AND WHERE
050  *         A IS MXN
060  *         B IS NXM
070  *       IT ALSO ZEROES UNUSED BOUDARY ELEMENTS OF MATRIX B
080        COMPLEX A(NDIM,NDIM),B(NDIM,NDIM)
090        IF(MAX0(M,N).GT.NDIM)CALL ERRSTP('CMTRAN',1)
100        DO 200 I=1,NDIM
110        DO 200 J=1,NDIM
120        IF(I.GT.M.OR.J.GT.N)GOTO100
130        B(J,I)=A(I,J)
140        GOTO200
150    100 B(J,I)=(0.,0.)
160    200 CONTINUE
170        RETURN
180        END
```

```
**********************************************************************

CMTSUBS1                                          04/08/81     3:56 PM

**********************************************************************


010**RUN *=;CMTSUB01(NOGO,CORE=30K)
020         SUBROUTINE CMTSUB(A,B,C,N,M)
030 *         THIS SUBTRACT ROUTINE PERFORMS THE COMPLEX MATRIX SUBTRACT
040 *         C=A-B USING THE UPPER LEFT NXN PARTITION OF MATRICES
050 *         A,B,C WHICH ARE EACH DIMENSIOD MXM
060 *         NOTE THIS ROUTINE CAN BE DONE IN PLACE, IE A=A-B
070         COMPLEX A(M,M),B(M,M),C(M,M)
080         IF(N.GT.M)CALL ERRSTR('CMTSUB',1)
090         DO 100 I=1,N
100         DO 100 J=1,N
110     100 C(I,J)=A(I,J)-B(I,J)
120         RETURN
130         END
```

```
CMTTMPS1                                                    04/08/81     3:57 PM

010**RUN *=;CMTTMPC1(NCC=,COPE=30K)
020        SUBROUTINE CMTTMP(A,B,C,L,M,N,NDIM)
030 *        THIS SUBROUTINE PERFORMS THE COMPLEX MATRIX TRANSPOSE
040 *        MULTIPLY C=AT*B WITH EACH DIMENSIONED MXM AND WHERE
050 *          A IS MXL
060 *          B IS MXN
070 *          C IS LXN
080        COMPLEX A(NDIM,NDIM),B(NDIM,NDIM),C(NDIM,NDIM),SUM,ZERO
090        DATA ZERO/(0.,0.)/
100        IF(N.GT.NDIM)CALL ERRSTP('CMTTMP',1)
110        DO 200 I=1,L
120        DO 200 J=1,N
130        SUM=ZERO
140        DO 100 K=1,M
150    100 SUM=SUM+A(K,I)*B(K,J)
160    200 C(I,J)=SUM
170        RETURN
180        END
```

```
CMTZROS1                                                    04/08/81     3:57 PM

010*#RUN *=:CMTZRO01(NOGO,CORE=30K)
020        SUBROUTINE CMTZRO(A,M,N,NDIM)
030 *        THIS SUBROUTINE ZEROES THE BORDER ELEMENTS OF COMPLEX
040 *        MATRIX A WHICH IS DIMENSIONED NDIMXNDIM
050        COMPLEX A(NDIM,NDIM),ZERO
060        DATA ZERO/(0.,0.)/
070        IF(M.GT.NDIM.OR.N.GT.NDIM)CALL ERRSTP('CMTZRO',1)
080        DO 100 I=1,M
090        DO 100 J=N+1,NDIM
100    100 A(I,J)=ZERO
110        DO 200 I=M+1,NDIM
120        DO 200 J=1,NDIM
130    200 A(I,J)=ZERO
140        RETURN
150        END
```

****

CMVMPYS1 04/08/81 3:57 PM

****

```
010**RUN *=;CMV' PYD1(NOGO,CORE=30K)
020       SUBROUTI E CMVMPY(A,V,W,L,M,LDIM,MDIM,NDIM)
030 *       THIS SUBROUTINE PERFORMS THE COMPLEX MATRIX-VECTOR MULTIPLY
040 *       W=A*V
050 *         A IS LXM
060 *         V IS MX1
070 *         W IS LX1
080 *       AND A,V,W ARE DIMENSIONED
090 *         A(LDIM,LDIM)
100 *         V(MDIM)
110 *         W(NDIM)
120 *       IN THE CALLI G PROGRA
130       COMPLEX A(LDIM,LDIM),V( DIM),W( DIM),ZERO,SUM
140       DATA ZERO/(0.,0.)/
150       IF(MAX0(L,M).GT.LDIM.OR.L.GT.MDIM.OR.M.GT.MDIM)
160      &  CALL ERRSTP('CMVMPY',1)
170       DO 200 I=1,L
180       SUM=ZERO
190       DO 100 J=1,M
200   100 SUM=SUM+A(I,J)*V(J)
210   200 W(I)=SUM
220       RETURN
230       END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
CSYADJS1                                                    04/08/81    3:57 PM

010*#RUN *=:CSYADJ01(NOGO,CORE=30K)
020       SUBROUTINE CSYADJ(A,N,M,IFSY,LU)
030 *       THIS SUBROUTINE FORCES THE MXM UPPER LEFT PARTITION OF AN MXM
040 *       COMPLEX MATRIX A TO BE SYMMETRIC BY USING THE AVERAGE
050 *       OF THE CORRESPONDING TO THE ELEMENT PAIRS
060 *       IF ENABLED BY IFSY.NE.0 A TEST IS MADE ON THEIR DIFFERENCE
070 *       AND IF ANY IS JUDGED TOO LARGE THEN ARRAY IS PRINTED OR
080 *       WRITTEN
090 *         IFSY.EQ.0 TESTING IS BYPASSED
100 *         IFSY.NE.0 TESTING IS PERFORMED
110 *           LU.EQ.06 FOR PRINT
120 *           LU.NE.06 FOR WRITE TO FILE    -
130 *        NOTE THIS PROCEDURE STOPS ON TEST FAIL
140       COMPLEX A(N,M),HALF,Z1,Z2,Z3
150       DATA HALF/(.5,0.)/,EPSI/1.E-3/
160       IF(N.GT.M)CALL ERRSTP('CSYADJ',0)
170       NM1=N-1
180       IF(IFSY.EQ.0)GOTO300
190       DO 200 I=1,NM1
200       IP1=I+1
210       DO 200 J=IP1,N
220       Z1=A(I,J)
230       Z2=A(J,I)
240       Z3=HALF*(Z1+Z2)
250 *       THE FOLLOWING TEST IS SOMEWHAT ARBITRARY
260       X=CABS(Z1-Z2)/CABS(Z3)
270       IF(X.GT.EPSI)GOTO500
280       A(I,J)=Z3
290   200 A(J,I)=Z3
300       RETURN
310   300 DO 400 I=1,NM1
320       IP1=I+1
330       DO 400 J=IP1,N
340       Z3=HALF*(A(I,J)+A(J,I))
350       A(I,J)=Z3
360   400 A(J,I)=Z3
370       RETURN
380   500 CALL CMTPRT(A,N,M,LU)
390       CALL ERRSTP('CTRADJ',1)
400       RETURN
410       END
```

```
010**RUN *=$CVDBAN01(NOGO,CORE=30K)
020       SUBROUTINE CVDBAN(V,N,M,NW,LU,WKAREA)
030 *       THIS SUBROUTINE PRINTS OR WRITES FOR DIAGNOSTIC PURPOSES
040 *       THE ELEMENTS OF A COMPLEX VECTOR IN POLAR DB FORM
050 *         LU.EQ.06 FOR PRINT
060 *         LU.NE.06 FOR WRITE TO FILE
070     11 FORMAT(9F9.2)
080     12 FORMAT(' ')
090        COMPLEX V(M),Z
100        DIMENSION WKAREA(NW)
110        DATA DEG/57.295780/
120        IF(N.GT.M.OR.(N+N).GT.NW)CALL ERRSTR('CVDBAN',1)
130        DO 200 I=1,N
140        Z=V(I)
150        A=CABS(Z)
160        IF(A.GT.(0.))GOTO170
170        WKAREA(I)=-999.99
180        WKAREA(I+N)=0.
190        GOTO200
200    170 WKAREA(I)=20.*ALOG10(A)
210        WKAREA(I+N)=DEG*ATAN2(AIMAG(Z),REAL(Z))
220    200 CONTINUE
230        WRITE(LU,11)(WKAREA(I),I=1,N)
240        WRITE(LU,11)(WKAREA(I+N),I=1,N)
250        WRITE(LU,12)
260        RETURN
270        END
```

CVECMPS1 04/08/81 3:57 PM

```
010**RUN *=:CVECMP01(NOGO,CORE=30K)
020       SUBROUTINE CVECMP(A,R,V,N,NDIM,ID)
030 *       THIS SUBROUTINE COMPARES AN INPUT COMPLEX VECTOR A
040 *       WITH A REFERENCE COMPLEX VECTOR R WHOSE ELEMENT NUMERICAL
050 *       VALUES ARE ESTABLISHED BY THE INITIAL EXECUTION
060 *       OF THIS SUBROUTINE
070     10 FORMAT(' ')
080     11 FORMAT(8F15.6)
090     12 FORMAT(A43,F15.6)
100        COMPLEX ZERO,TEMP
110        COMPLEX A(NDIM),R(NDIM),V(NDIM)
120        DATA IFLAG/0/,RZERO/0./,ZERO/(0.,0.)/
130        IF(IFLAG.EQ.1)GOTO200
140        DO 120 I=1,N
150   120  R(I)=A(I)
160        IFLAG=1
170        IF(N.EQ.NDIM)GOTO200
180        DO 150 I=N+1,NDIM
190   150  R(I)=RZERO
200   200  CONTINUE
210        XMAX=RZERO
220        DO 220 I=1,N
230        X=CABS(R(I))
240        IF(X.EQ.RZERO)GOTO210
250        TEMP=(A(I)-R(I))/X
260        V(I)=TEMP
270        X=CABS(TEMP)
280        IF(X.GT.XMAX)XMAX=X
290        GOTO220
300   210  V(I)=ZERO
310   220  CONTINUE
320        CALL FORMFE
330        PRINT,'ID=',ID
340        PRINT,'PERTURBED VECTOR'
350        CALL CVEPRT(A,N,NDIM,06)
360        PRINT 10
370        PRINT,'REFERENCE VECTOR'
380        CALL CVEPRT(R,N,NDIM,06)
390        PRINT 10
400        PRINT,'NORMALIZED DIFFERENCE VECTOR'
410        PRINT 10
420        PRINT 11,( REAL(V(I)),I=1,N)
430        PRINT 11,(AIMAG(V(I)),I=1,N)
440        PRINT 10
450        PRINT 12,'MAXIMUM NORMALIZED DIFFERENCE ELEMENT=',XMAX
460        CALL FORMFE
470        RETURN
480        END
```

```
010**RUN *=;CVEPOLO1(NOGO,CORE=30K)
020       SUBROUTINE CVEPOL(V,N,M,LU,IW,WKAREA)
030 *       THIS SUBROUTINE PRINTS OR WRITES FOR DIAGNOSTIC PURPOSES
040 *       THE ELEMENTS OF A COMPLEX VECTOR IN POLAR FORM
050 *         LU.EQ.06 FOR PRINT
060 *         LU.NE.06 FOR WRITE TO FILE
070 *       NOTE IF N.GT.8 PROBABLY BEST TO REDUCE 15 FIELD WIDTH
080    11 FORMAT(1P8E15.7)
090    13 FORMAT(9F15.7)
100       COMPLEX V(M),Z
110       DIMENSION WKAREA(NW)
120       DATA DEG/57.29578/,ZERO/0./
130       IF(N.GT.M.OR.(N+N).GT.NW)CALL EPPSTP('CVEPOL',1)
140       DO 200 I=1,N
150       Z=V(I)
160       X=REAL(Z)
170       Y=AIMAG(Z)
180       WKAREA(I)=CABS(Z)
190       IF(ABS(X).GT.ZERO)GOTO190
200       IF(ABS(Y).GT.ZERO)GOTO190
210       WKAREA(I+N)=ZERO
220       GOTO200
230   190 WKAREA(I+N)=DEG*ATAN2(Y,X)
240   200 CONTINUE
250       WRITE(LU,11)(WKAREA(I),I=1,N)
260       WRITE(LU,13)(WKAREA(I+N),I=1,N)
270       RETURN
280       END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
010**RUN *=:CVEPRTO1(:OGO.CORE=30K)
020       SUBROUTINE CVEPRT(V.N.M.LU)
030 *       THIS SUBROUTINE PRINTS OR WRITES FOR DIAGNOSTIC PURPOSES
040 *       THE ELEMENTS OF A COMPLEX VECTOR
050 *         LU.EQ.06 FOR PRINT
060 *         LU.NE.06 FOR WRITE TO FILE
070 *       NOTE IF M.GT.8, PROBABLY BEST TO REDUCE 15 FIELD WIDTH
080     11 FORMAT(1P8E15.7)
090     12 FORMAT(' ')
100        COMPLEX V(M)
110        IF(N.GT.M)CALL ERRSTP('CVEPRT',1)
120        WRITE(LU,11)( REAL(V(J)),J=1,N)
130        WRITE(LU,11)(AIMAG(V(J)),J=1,N)
140        WRITE(LU,12)
150        RETURN
160        END
```

```
CZYREDS1                                                    04/08/81    3:58 PM

010*#RUN *=;CZYRED01(NOGO,CORE=30K)
020        SUBROUTINE CZYRED(Z,Y,ZNECO,YNECO,N,NECO,M,NW,WKAREA)
030 *        THIS SUBROUTINE PERFORMS THE MATRIX ORDER REDUCTION OF
040 *        THE N CONDUCTOR Z AND Y MATRICES (WHICH ARE THE UPPER
050 *        NXN PARTITION OF MXM RESPECTIVE ARRAYS) TO NECO EFFECTIVE
060 *        CONDUCTORS BASED ON THE ASSUMPTION THAT N-NECO CONDUCTORS
070 *        ARE AT ZERO POTENTIAL.  THE RESULTING REDUCED MATRICES
080 *        ARE PLACED IN THE UPPER LEFT NECOXNECO PARTITION OF ARRAYS
090 *        ZNECO,YNECO EACH DIMENSIOD MXM  FOR PROGRAMMING-
100 *        CONVENIENCE AND SINCE THIS ROUTINE USED ONCE AND FOR ALL,
110 *        WILL USE FULL MATRIX INVERSION ALTERNATIVE.  SEE RCR APPENDIX A
120        COMPLEX Z(M,M),Y(M,M),ZNECO(M,M),YNECO(M,M)
130        IF(N.GT.M)CALL ERRSTP('CZYRED',1)
140        CALL CMTCOP(Z,ZNECO,M,M)
150        CALL CMTCOP(Y,YNECO,M,M)
160 *        NOTE THAT SINCE THE UPPER NECO*NECO PARTITION REPRESENTS
170 *        YNECO, JUST COPY WHOLE N*N Y MATRIX
180        IF(N.EQ.NECO)RETURN
190        CALL CMTINV(ZNECO,N,M,1,WKAREA)
200        CALL CMTINV(ZNECO,NECO,NECO,M,WKAREA)
210        RETURN
220        END
```

```
010**RUN *=:DPUFRW02(NOGO,CORE=30K)
020       SUBROUTINE DPUFRW(LU,IR,IRW,M,LTYP,NECO,NTOT,NUMF,NUMS,NEUS,NEUF,
030      &                  Z,Y,ZO,YO,S,SI,DL)
040 *       THIS SUBROUTINE MAKES CONVENIENT THE WRITING TO AND READING
050 *       FROM DATA RECORD ASSOCIATED WITH A GIVEN LTYP IN
060 *       DATAFILE DPULIJKN WHICH HAS BEEN PREVIOUSLY ASSIGNED A FRN
070 *       LU.  THE CONVENIENCE REFERS TO THE ABILITY TO WRITE/READ
080 *       THE UPPER LEFT NECOXNECO PARTITIONS OF MATRICES
090 *       DIMENSIONED MXM
100 *       NOTE THAT ON READ THE COMPLEX CONTROL PARAMETER NECO IS
110 *       OBTAINED FROM THE FIRST PART OF THE READ AND IS A RETURNED
120 *       VARIABLE VALUE.  ON WRITE NECO MUST BE PROVIDED.
130 *       THE DIMENSION M MUST BE THAT USED IN THE CALLING
140 *       PROGRAM FOR MATRICES
150 *       IRW.NE.1 FOR READ
160 *       IRW.EQ.1 FOR WRITE
170       COMPLEX Z(M,M),ZO(M,M),S(M,M),SI(M,M),DL(M)
180       COMPLEX Y(M,M),YO(M,M)
190       IF(IR.LE.2)CALL ERRSTT('DPUFRW',IR,IRW,M,NECO)
200       IF(IRW.EQ.1)GOTO100
210       READ(LU'IR)LTYP,NECO,NTOT,NUMF,NUMS,NEUS,NEUF,
220      &(( Z(I,J),I=1,NECO),J=1,NECO),
230      &(( Y(I,J),I=1,NECO),J=1,NECO),
240      &((ZO(I,J),I=1,NECO),J=1,NECO),
250      &((YO(I,J),I=1,NECO),J=1,NECO),
260      &(( S(I,J),I=1,NECO),J=1,NECO),
270      &((SI(I,J),I=1,NECO),J=1,NECO),
280      &(DL(I),I=1,NECO)
290       RETURN
300   100 WRITE(LU'IR)LTYP,NECO,NTOT,NUMF,NUMS,NEUS,NEUF,
310      &(( Z(I,J),I=1,NECO),J=1,NECO),
320      &(( Y(I,J),I=1,NECO),J=1,NECO),
330      &((ZO(I,J),I=1,NECO),J=1,NECO),
340      &((YO(I,J),I=1,NECO),J=1,NECO),
350      &(( S(I,J),I=1,NECO),J=1,NECO),
360      &((SI(I,J),I=1,NECO),J=1,NECO),
370      &(DL(I),I=1,NECO)
380       RETURN
390       END
```

```
**********************************************************************

ERRSTPS1     ,                                    04/08/81     3:58 PM

**********************************************************************


010*#RUN *$:ERRSTP01(NOGO,CORE=30K)
020        SUBROUTINE ERRSTP(NAME,NCODE)
030 *        THIS GENERAL STOP ROUTINE PRINTS THE SIX CHARACTER
040 *        NAME AND INTEGER CODE NCODE
050        CHARACTER NAME*6
060     10 FORMAT('STOPPED IN ',A6,' NCODE=',I6)
070        PRINT 10,NAME,NCODE
080        STOP 9999
090        END
```

```
**********************************************************************

ERRSTTS1                                          04/08/81    3:58 PM

**********************************************************************


010**RUN *=:ERRSTTO1(NOGO,CORE=30K)
020         SUBROUTINE ERRSTT(NAME,NCODE,I1,I2,I3)
030 *         THIS GENERAL STOP ROUTINE PRINTS THE SIX CHARACTER
040 *         NAME AND INTEGER CODE NCODE
050         CHARACTER NAME*6
060      10 FORMAT('STOPPED IN ',A6,' NCODE=',I6,3I11)
070         PRINT 10,NAME,NCODE,I1,I2,I3
080         STOP 9999
090         END
```

```
FILSTPS1                                                    04/08/81     3:59 PM

010         SUBROUTINE FILSTOP(NFILE,IQUAN,IN,JN,NCHECK)
020     *   ON FILE READ THIS SUBROUTINE PROVIDES A GENERAL STOP UPON
030     *   ERROR DETECTION BETWEEN DATA IN FILE AND EXPECTED VALUES.
040     *   THIS SUBROUTINE WILL BE USED IN COMPARING HEADER INFORMATION
050     *   WITH OTHER NETWORK TO ASSURE THAT THE PROPER FILE HAS BEEN
060     *   ACCESSED.
070         CHARACTER*9 NFILE, IQUAN
080         PRINT 10, NFILE, IQUAN, IN,JN
090         10   FORMAT("STOPPED FOR READ ERROR ON FILE ",A9/"VARIABLE  ",A9,
100      &  "   SHOULD BE   ",I8,"  BUT QUANTITY READ IS  ",I8)
110         STOP NCHECK
120         END
```

```
OPNVOLS1                                                    04/08/81     3:59 PM

010**RUN *=$OPNVOLO1(NOGO,CORE=30K)
020       SUBROUTINE OPNVOL(Y,E,S,V,N,M,NDIM,WKAREA,NW)
030 *       THIS SUBROUTINE COMPUTES AND FORMS THE TOTAL N ELEMENT
040 *       VOLTAGE VECTOR E, GIVEN THAT THE LAST N-M CURRENT ELEMENTS
050 *       OF THE N ELEMENT CURRENT VECTOR ARE ZERO, IE OPEN
060 *       CIRCUIT, AND GIVEN THE FIRST M ELEMENTS OF THE N ELEMENT
070 *       VOLTAGE VECTOR E, AND THE NXN ADMITTANCE MATRIX Y.  ALL
080 *       VECTORS AND MATRICES ARE ASSUMED DIMENSIONED NDIM OR
090 *       NDIMXNDIM IN CALLING PROGRAM.
100 *       THE COMPUTATION FOLLOWS THE MODEL
110 *
120 *         I1=Y11*E1+Y12*E2        E2=-Y22I*Y21*E1
130 *
140 *          0=Y21*E1+Y22*E2        I1=(Y11-Y12*Y22I*Y21)*E1
150 *
160 *       NOTE S AND V ARE SCRATCH
170       COMPLEX Y(NDIM,NDIM),E(NDIM),ZERO
180       COMPLEX S(NDIM,NDIM),V(NDIM),ZSUM
190       DIMENSION WKAREA(NW)
200       DATA ZERO/(0.,0.)/
210       IF(N.GT.NDIM.OR.M.GE.N)CALL ERRSTP('OPNVOL',0)
220       DO 200 K=1,N-M
230       I=K+M
240 *       FIRST FORM V=Y21*E1
250       ZSUM=ZERO
260       DO 100 J=1,M
270   100 ZSUM=ZSUM+Y(I,J)*E(J)
280       V(K)=ZSUM
290 *       NEXT FORM S=Y22
300       DO 200 L=1,N-M
310       J=L+M
320       S(K,L)=Y(I,J)
330   200 CONTINUE
340       J=N-M
350       CALL CMTINV(S,J,J,NDIM,WKAREA)
360 *       NOW CALCULATE E2.   NOTE S=Y22I, V=Y21*E1
370       DO 400 I=1,N-M
380       ZSUM=ZERO
390       DO 300 J=1,N-M
400   300 ZSUM=ZSUM-S(I,J)*V(J)
410   400 E(I+M)=ZSUM
420       RETURN
430       END
```

```
OPTADMS1                                                    04/08/81     3:59 PM

010*#RUN *=:OPTADM01(NCGO,CORE=30K)
020       SUBROUTINE OPTADM(A,B,T,M,N,NDIM,WKAREA,NW)
030 *       THIS SUBROUTINE DETERMINES THE 'MX' DRIVING POINT ADMITTANCE
040 *       MATRIX B, AND THE N-MXM VOLTAGE TRANSFER MATRIX T ASSOCIATED
050 *       WITH A NXN ADMITTANCE MATRIX A WHICH HAS ITS LAST N-M TERMINALS
060 *       OPEN CIRCUITED.  ALL MATRICES ARE ASSUMED DIMENSIONED NDIMXNDIM
070 *       IN THE CALLING PORGRAM.
080 *         THE COMPUTATION IS BASED ON THE FOLLOWING PARTITION OF A
090 *
100 *                    M    N-M
110 *
120 *              M    Y11  Y12          T=-Y22I*Y21
130 *
140 *            N-M    Y21  Y22          B=Y11-Y12*Y22I*Y21=Y11+Y12*T
150 *
160       COMPLEX A(NDIM,NDIM),B(NDIM,NDIM),T(NDIM,NDIM)
170       COMPLEX SUM,ZERO
180       DIMENSION WKAREA(NW)
190       DATA ZERO/(0.,0.)/
200       IF(N.GT.NDIM.OR.M.GE.N)CALL ERRSTT('OPTADM',0,M,N,NDIM)
210 *       FIRST USE B TEMPORARILY TO OPTAIN B=Y22I
220       DO 100 K=1,N-M
230       I=K+M
240       DO 100 L=1,N-M
250       J=L+M
260       B(K,L)=A(I,J)
270   100 CONTINUE
280       L=N-M
290       CALL CMTINV(B,L,L,NDIM,WKAREA)
300 *       NEXT FORM T=-Y22I*Y21
310       DO 300 I=1,L
320       DO 300 J=1,M
330       SUM=ZERO
340       DO 200 K=1,L
350   200 SUM=SUM-B(I,K)*A(K+M,J)
360   300 T(I,J)=SUM
370 *       NEXT FORM B=Y11+Y12*T
380       DO 500 I=1,M
390       DO 500 J=1,M
400       SUM=A(I,J)
410       DO 400 K=1,L
420   400 SUM=SUM+A(I,K+M)*T(K,J)
430   500 B(I,J)=SUM
440       RETURN
450       END
```

```
010**RUN *=;PROSEQS2(LOGO,CORE=30K)
020       SUBROUTINE PROSEQ(INLCSA,INSEQV,ISTAK,NMAX ,N,M,JROOT)
030 *      THE PURPOSE OF THIS SUBROUTINE IS TO CONSTRUCT THE
040 *      PROCESSING SEQUENCE VECTOR INSEQV USING THE NODE
050 *      LOGICAL CONNECTION AND STATUS ARRAY INLCSA. THE
060 *      SUBROUTINE ALSO COMPUTES THE LEVEL OF EACH NODE IN
070 *      THE BINARY TREE WITH THE NODE CONNECTED TO THE
080 *      SOURCE BEING AT LEVEL ZERO. THE LEVEL ONLY CHANGES
090 *      ON EACH BRANCH. ISTAK IS A TEMPORARY STACK VECTOR
100 *        INLCSA (I,1) = NODE INDEX (REDUNDANT CHECK)
110 *        INLCSA (I,2) = WHERE FROM NODE (PARENT NODE)-IPAR
120 *        INLCSA (I,3) = WHERE TO NODE 1 - LEFTSON
130 *        INLCSA (I,4) = WHERE TO NODE 2 - RIGHTSON
140 *        INLCSA (I,5) = LEVEL OF NODE
150 *
160 *        IN IS USED AS AN INDEX WHICH FILLS INSEQV FROM
170 *        TOP DOWN
180 *      N  IS A DUMMY VARIABLE ON  INPUT WHICH RETURNS NUMBER OF NODES
190 *          JROOT IS FIRST NODE FROM SOURCE
200 *          I = STACK INDEX
210       DIMENSION INLCSA(NMAX,M),INSEQV(NMAX),ISTAK(NMAX)
220       DO  25  I = 1 , NMAX
230    25 INLCSA(I,5) =  -10
240       I = 0
250       NODE = JROOT
260       INLCSA(NODE,5) = 0
270       INSEQV(1) = NODE
280       IN =  2
290     1 IF (INLCSA(NODE,3).EQ.0) GO TO 2
300       I = I + 1
310       ISTAK(I) = INLCSA(NODE,3)
320     2 NODE = INLCSA(NODE,4)
330       IF(NODE.NE.0) GO TO 3
340       N  =  IN - 1
350       IF (I.EQ.0) RETURN
360       NODE = ISTAK (I)
370       I = I-1
380     3 INSEQV(IN) = NODE
390       IF (INLCSA(NODE,1) .LT. 0)  GO TO 23
400       IN = IN+1
410       IPAR = INLCSA (NODE,2)
420       IF(INLCSA(IPAR,5) .LT. 0)  GO TO 20
430       IS = MIN0(INLCSA(IPAR,3),INLCSA(IPAR,4))
440       IF (IS .EQ. 0) GO TO 4
450       INLCSA(NODE,5) = INLCSA(IPAR,5) + 1
460       GO TO 1
470     4 INLCSA(NODE,5) = INLCSA (IPAR,5)
480       GO TO 1
490    20 WRITE(42,1001) IPAR
```

```
500 1001  FORMAT("NETWORK CONFIGURATION ERROR"/"NODE ",I6,"NOT DEFINED"/"PROGR
510       STOP
520    23 WRITE(*,1002) NODE
530 1002  FORMAT("NODE NUMBER ",I5," IS NOT DEFINED"/"PROGRAM ABORTS")
540       STOP
550       END
```

```
RVEPRTS1                                                    04/08/81     3:59 PM

010**RUN *=;RVEPRT01(NOGO,CORE=30K)
020        SUBROUTINE RVEPRT(V,N,M,LU)
030 *        THIS SUBROUTINE PRINTS OR WRITES FOR DIAGNOSTIC PURPOSES
040 *        THE FIRST N ELEMENTS OF A REAL VECTOR WITH DIMENSION M
050 *          LU.EQ.06 FOR PRINT
060 *          LU.NE.06 FOR WRITE TO FILE
070 *        NOTE IF N.GT.9 PROBABLY BEST TO REDUCE 13 FIELD WIDTH
080     11 FORMAT(1P9E13.5)
090        DIMENSION V(M)
100        IF(N.GT.M)CALL ERRSTP('RVEPRT',1)
110        WRITE(LU,11)(V(J),J=1,N)
120        RETURN
130        END
```

```
**********************************************************************

TRANADS4                                          04/08/81     4:00 PM

**********************************************************************


010*#RUN *=;TRANADS4(NOGO,CORE=30K)
020       SUBROUTINE TRANAD(Y,IFRE,ITTY,LUTR,TRANFI,TRANAY,NDIM,MDIM)
030 *       THIS VERSION OF TRANAD READS A REVISED STRUCTURE FOR THE
040 *       TRANSFORMER FILE TRANFI.  THE NEW STRUCTURE CONSISTS OF
050 *       NON-LINE NUMBERED RECORDS EACH WITH THE FORMAT
060 *         IFRE ITTY AM TH
070 *       WHERE
080 *         IFRE=TWO DIGIT FREQUENCY CODE
090 *         ITTY=ONE,TWO,THREE DIGIT TRANSFORMER TYPE CODE
100 *         AM  =ADMITTANCE AMPLITUDE IN DB
110 *         TH  =ADMITTANCE ANGLE IN DEGREES
120 *       THE AM FIELD SHOULD BEGIN IN COLUMN 9 OR GREATER TO FACILITATE
130 *       TOPS EDITOR SORTING (WHICH IS FOR USER CONVENIENCE ONLY)
140 *       THE COMPLEX MATRIX TRANAY(L,K) WILL CONTAIN COMPLEX SCALAR
150 *       PHASOR ADMITTANCES FOR EACH (DEFINED) COMBINATION OF IFREQ
160 *       AND ITTY.  FOR ALL UNDEFINED CASES THE VALUE YBAD WILL BE
170 *       USED AND A WARNING PRINTED TO TERMINAL FIVE TIMES MAXIMUM.
180 *       THE COLUMNS OF TRANAY WILL CORRESPOND TO (THRU A DIRECTORY
190 *       VECTOR JCOL) ITTY.  SIMILARLY THE ROWS OF TRANAY WILL
200 *       CORRESPOND TO (THRU A DIRECTORY VECTOR IROW) IFREQ.
210 *
220 *       DIRECTORY VECTORS JCOL AND IROW ARE USED TO SAVE CORE. I.E.
230 *       ANY ONE USER WILL BE INTERESTED IN ONLY RELATIVELY SMALL
240 *       NUMBER OF DISTINCT FREQUENCY CODES.  SIMILARLY FOR ITTY
250 *       SO THAT TRANAY CAN BE DIMENSIONED MUCH SMALLER USING DIRECTORY
260 *       VECTORS THAN USING DIRECT ADDRESSING VIA IFREQ AND ITTY
270 *
280 *       DIRECTORY VECTORS ARE DIMENSIONED SUCH THAT
290 *         IFRE.LE.IMAX
300 *         ITTY.LE.JMAX
310     10 FORMAT(V)
320        COMPLEX Y,TRANAY(NDIM,MDIM),YBAD
330        DIMENSION IROW(99),JCOL(167)
340        CHARACTER NAME*6,TRANFI*9
350        DATA RAD/1.7453293E-2/,IFLG/1/,NAME/'TRANAD'/
360        DATA IMAX/99/,JMAX/167/,NCOUNT/1/,YBAD/(1.E-30,1.E-30)/
370        IF(IFLG.EQ.0)GOTO200
380 *        ON FIRST PASS SETUP DIRECTORY VECTORS AND TRANAY
390        IFLG=0
400        CALL OPENF(LUTR,TRANFI,ISTAT,1,0,1)
410        IF(ISTAT.EQ.0)GOTO100
420        PRINT,'UNABLE TO OPEN FILE=',TRANFI
430        CALL ERRSTT(NAME,0,ISTAT,IFRE,ITTY)
440    100 CONTINUE
450        DO 102 I=1,IMAX
460    102 IROW(I)=0
470        DO 104 J=1,JMAX
480    104 JCOL(J)=0
490        DO 106 I=1,NDIM
```

```
500         DO 106 J=1,NDIM
510     106 TRANAY(I,J)=YBAD
520         NROW=1
530         NCOL=1
540     120 READ(LUTR,10,END=150)I,J,AM,TH
550         IF(I.GT.IMAX.OR.J.GT.JMAX)CALL ERRSTT(NAME,1,I,J,JMAX)
560         IF(IROW(I).GT.0)GOTO130
570         IROW(I)=NROW
580         IF(NROW.GT.NDIM)CALL ERRSTT(NAME,2,NROW,NDIM,I)
590         NROW=NROW+1
600     130 CONTINUE
610         IF(JCOL(J).GT.0)GOTO140
620         JCOL(J)=NCOL
630         IF(NCOL.GT.NDIM)CALL ERRSTT(NAME,3,NCOL,NDIM,J)
640         NCOL=NCOL+1
650     140 CONTINUE
660         AM=10.**(.05*AM)
670         TH=RAD*TH
680         TRANAY(IROW(I),JCOL(J))=CMPLX(AM*COS(TH),AM*SIN(TH))
690         GOTO120
700     150 CALL DETACH(LUTR,ISTAT,)
710     200 CONTINUE
720         IF(IFRE.GT.IMAX.OR.ITTY.GT.JMAX)CALL ERRSTT(NAME,4,IFRE,ITTY,JMAX)
730         NROW=IROW(IFRE)
740         NCOL=JCOL(ITTY)
750         IF(NROW.EQ.0.OR.NCOL.EQ.0)CALL ERRSTT(NAME,5,IFRE,ITTY,NCOL)
760         Y=TRANAY(NROW,NCOL)
770         IF(Y.NE.YBAD.OR.NCOUNT.GT.5)RETURN
780         PRINT,'WARNING, USING UNDEFINED TRANSFORMER DATA'
790         NCOUNT=NCOUNT+1
800         RETURN
810         END
```

```
010**RUN *=;YADDECS1(NOGO,CORE=30K)
020       SUBROUTINE YADDEC(Y,N,M,LOUT)
030 *       THIS SUBROUTINE DECOMPOSES A 3X3 OR 4,4 ANALYTIC MATRIX
040 *       INTO DISCRETE PHYSICAL ELEMENTS. THE CONDUCTORS ARE DESIGNATED
050 *       1,2,3,N AND FOR THE 4X4 CASE THE GROUND IS DENOTED BY G
060 *       FOR THE 3X3 DELTA CONNECTED SYSTEM ADMITTANCE Y1N,Y2N,Y3N ARE
070 *       TO BE INTERPRETED AS Y1G,Y2G,Y3G
080 *
090 *       NOTE THAT FOR THE 3X3 CASE ONLY UPPER TRIANGE ELEMENTS
100 *       ARE USED.  ARBITRARILY FOR CODING CONVENIENCE, ALL ELEMENTS
110 *       OF Y ARE USED IN THE 4X4 CASE.  NATURALLY IF Y IS EXACTLY
120 *       SYMMETRIC NO UNCERTAINITIES ARE INVOLVED.
130       COMPLEX Y(M,M),YSUM(4),ZERO,SUM
140       DATA ZERO/(0.,0.)/
150       IF(N.GT.M.OR.N.GT.4.OR.N.LT.3)CALL ERRSTP('YADDEC',N)
160    10 FORMAT(1P6E15.7)
170    11 FORMAT(6X,'Y1N',12X,'Y12',12X,'Y13',12X,'Y2N',12X,'Y23',12X,'Y3N')
180    12 FORMAT(6X,'Y1G',12X,'Y2G',,12X,'Y3G',12X,'YNG')
190    13 FORMAT(' ')
200       CALL FPARAM(1,120)
210       WRITE(LOUT,11)
220       IF(N.EQ.4)GOTO200
230       WRITE(LOUT,10)REAL(Y(1,1)+Y(1,2)+Y(1,3)),-REAL(Y(1,2)),
240      &              -REAL(Y(1,3)),REAL(Y(1,2)+Y(2,2)+Y(2,3)),
250      &              -REAL(Y(2,3)),REAL(Y(1,3)+Y(2,3)+Y(3,3))
260       WRITE(LOUT,10)AIMAG(Y(1,1)+Y(1,2)+Y(1,3)),-AIMAG(Y(1,2)),
270      &              -AIMAG(Y(1,3)),AIMAG(Y(1,2)+Y(2,2)+Y(2,3)),
280      &              -AIMAG(Y(2,3)),AIMAG(Y(1,3)+Y(2,3)+Y(3,3))
290       WRITE(LOUT,13)
300       RETURN
310   200 CONTINUE
320       WRITE(LOUT,10)-REAL(Y(1,4)),-REAL(Y(1,2)),-REAL(Y(1,3)),
330      &               -REAL(Y(2,4)),-REAL(Y(2,3)),-REAL(Y(3,4))
340       WRITE(LOUT,10)-AIMAG(Y(1,4)),-AIMAG(Y(1,2)),-AIMAG(Y(1,3)),
350      &               -AIMAG(Y(2,4)),-AIMAG(Y(2,3)),-AIMAG(Y(3,4))
360       WRITE(LOUT,13)
370       DO 400 I=1,4
380       SUM=ZERO
390       DO 300 J=1,4
400   300 SUM=SUM+Y(I,J)
410   400 YSUM(I)=SUM
420       WRITE(LOUT,12)
430       WRITE(LOUT,10)( REAL(YSUM(I)),I=1,4)
440       WRITE(LOUT,10)(AIMAG(YSUM(I)),I=1,4)
450       WRITE(LOUT,13)
460       RETURN
470       END
```

```
YADMITS1                                                    04/08/81     4:00 PM

010**RUN *=;YADMIT01(NOGO,CORE=30K)
020       SUBROUTINE YADMIT(Y11,Y22,Y33,Y12,Y23,Y13,
030      &                  Y1N,Y2N,Y3N,YNN,Y,YR,N,M,IOPT,LU,IDIA)
040 *       THIS SUBROUTINE COMPUTES FOR IOPT=0, THE ANALYTIC COMPLEX
050 *       ADMITTANCE Y GIVEN THE PHYSICAL ELEMENTS, AND FOR IOPT=1
060 *       PERFORMS THE INVERSE OPERATION OF DECOMPOSING AN ANALYTIC
070 *       COMPLEX MATRIX INTO ITS PHYSICAL COMPONENTS.  ALSO FOR THIS
080 *       LATTER CASE CHECKS TO MAKE SURE EACH PHYSICAL COMPONENT
090 *       HAS POSITIVE REAL PART
100 *
110 *       THIS SUBROUTINE KEYS ON N=3 OR N=4 FOR CORRESPONDING LOGIC
120     10 FORMAT('ELEMENT HAS NEGATIVE REAL PART'/
130      &       'Y11,Y22,Y33,Y12,Y23,Y13,Y1N,Y2N,Y3N,YNN')
140     11 FORMAT(1P6E11.3)
150     12 FORMAT('REALIZABILITY STATUS= ',I3)
160        DIMENSION YR(M,M)
170        CHARACTER*6 NAME
180        COMPLEX Y(M,M),Y11,Y22,Y33,Y12,Y23,Y13,Y1N,Y2N,Y3N,YNN,ZERO
190        DATA ZERO/(0.,0.)/,RZERO/0./,NAME/'YADMIT'/
200        IF(N.LT.3.OR.N.GT.M.OR.N.GT.4)CALL ERRSTT(NAME,5,N,M,IOPT)
210        IF(IOPT.NE.0)GOTO200
220 *       PERFORM PHYSICAL TO ANALYTIC
230        Y(1,1)=Y11+Y12+Y13+Y1N
240        Y(2,2)=Y12+Y22+Y23+Y2N
250        Y(3,3)=Y13+Y23+Y33+Y3N
260        Y(1,2)=-Y12
270        Y(2,1)=-Y12
280        Y(1,3)=-Y13
290        Y(3,1)=-Y13
300        Y(2,3)=-Y23
310        Y(3,2)=-Y23
320        IF(N.EQ.3)RETURN
330        Y(1,4)=-Y11
340        Y(4,1)=-Y11
350        Y(2,4)=-Y22
360        Y(4,2)=-Y22
370        Y(3,4)=-Y33
380        Y(4,3)=-Y3_
390        Y(4,4)=Y11+Y22+Y33+YNN
400        RETURN
410     200 CONTINUE
420        CALL CSYADJ(Y,N,M,IDIA,LU)
430        CALL YREALI(Y,YR,N,M,LU,IDIA,ISTAT)
440        WRITE(LU,12)ISTAT
450 *       PERFORM ANALYTIC TO PHYSICAL
460        Y12=-Y(1,2)
470        Y13=-Y(1,3)
480        Y23=-Y(2,3)
490        Y11=Y(1,1)-Y12-Y13
```

```
      Y22=Y(2,2)-Y23-Y12
      Y33=Y(3,3)-Y13-Y23
      YNN=ZERO
      Y1N=ZERO
      Y2N=ZERO
      Y3N=ZERO
      IF(REAL(Y12).LT.RZERO)GOTO300
      IF(REAL(Y13).LT.RZERO)GOTO300
      IF(REAL(Y23).LT.RZERO)GOTO300
      IF(REAL(Y11).LT.RZERO)GOTO300
      IF(REAL(Y22).LT.RZERO)GOTO300
      IF(REAL(Y33).LT.RZERO)GOTO300
      IF(N.EQ.3)RETURN
      Y11=-Y(1,4)
      Y22=-Y(2,4)
      Y33=-Y(3,4)
      Y1N=Y(1,1)-Y12-Y13-Y11
      Y2N=Y(2,2)-Y23-Y12-Y22
      Y3N=Y(3,3)-Y13-Y23-Y33
      YNN=Y(4,4)-Y11-Y22-Y33
      IF(REAL(Y11).LT.RZERO)GOTO300
      IF(REAL(Y22).LT.RZERO)GOTO300
      IF(REAL(Y33).LT.RZERO)GOTO300
      IF(REAL(Y1N).LT.RZERO)GOTO300
      IF(REAL(Y2N).LT.RZERO)GOTO300
      IF(REAL(Y3N).LT.RZERO)GOTO300
      IF(REAL(YNN).LT.RZERO)GOTO300
      RETURN
  300 WRITE(LU,10)
      WRITE(LU,11)Y11,Y22,Y33,Y12,Y23,Y13,Y1N,Y2N,Y3N,YNN
      CALL ERRSTT(NAME,4,A,M,IOPT)
      STOP 0300
      END
```

```
YADMNUS1                                                    04/08/81    4:00 PM

010**RUN *=;YADMNUS1(NOGO,CORE=30K)
020       SUBROUTINE YADMNU(Y,YN,N,M)
030 *       THIS SUBROUTINE CONVERTS AN NXN COMPLEX MATRIX REPRESENTING
040 *       A PERFECTLY GROUNDED N TERMINAL ADMITTANCE INTO AN N+1XN+1
050 *       (N.LT.M) NON-PERFECTLY GROUNDED ADMITTANCE REPRESENTATION
060 *       ACCORDING TO FIGURE I-1 OF RCR APPENDIX I.
070 *       NOTE THAT FOR THIS SUBROUTINE, ANALYTIC ELEMENTS ARE USED
080       COMPLEX Y(M,M),YN,SUM,SUMN,ZERO
090       DATA ZERO/(0.,0.)/
100       IF(N.GE.1)CALL EPRSTT('YADMNU',0,N,M,0)
110       NP1=N+1
120       SUMN=ZERO
130       DO 200 I=1,N
140       SUM=ZERO
150       DO 100 J=1,N
160   100 SUM=SUM+Y(I,J)
170       Y(I,NP1)=-SUM
180       Y(NP1,I)=-SUM
190   200 SUMN=SUMN+SUM
200       Y(NP1,NP1)=SUMN+YN
210       RETURN
220       END
```

```
**********************************************************************

YREALIS1                                          04/08/81     4:01 PM

**********************************************************************


010**RUN *=;YREALID1(NOGO,CORE=30K)
020       SUBROUTINE YREALI(Y,YR,N,M,LU,IDIA,ISTAT)
030 *       THIS SUBROUTINE CHECKS A SPECIFIED COMPLEX ADMITTANCE Y
040 *       TO DETERMINE IF IT IS PHYSICALLY REALIZABLE  WITH
050 *       RESPECT TO HAVING A POSITIVE DEFINITE REAL
060 *       MATRIX.  THE METHOD OF PRICIPAL MINORS IS USED TO TEST
070 *       POSITIVE REAL.
080 *       NOTE THIS SUBROUTINE USES OLDTSLIB ROUTINE
090 *       DETE(YR,N,M) TO EVALUATE DETERMINANTS
100       COMPLEX Y(M,M)
110       DIMENSION YR(M,M)
120       DATA RZERO/0./
130    10 FORMAT('PRINCIPAL MINOR',I2,' =',1PE13.4)
140       IF(N.GT.M)CALL ERRSTP('YREALI',0)
150       ISTAT=0
160       DO 200 K=1,N
170       DO 100 I=1,N
180       DO 100 J=1,N
190   100 YR(I,J)=REAL(Y(I,J))
200       X=DETE(YR,K,M)
210       IF(X.GT.RZERO)GOTO200
220       ISTAT=1
230       IF(IDIA.NE.0)WRITE(LU,10)K,X
240   200 CONTINUE
250       RETURN
260       END
```

```
010**RUN *=;YTRAMTO2(NOGO,CORE=30K)
020        SUBROUTINE YTRANT(Y,YT,IPHSE,NDIM)
030 *        THIS SUBROUTINE EXPANDS A SCALAR COMPLEX TRANSFORMER
040 *        DRIVING POINT ADMITTANCE INTO AN ADMITTANCE MATRIX
050 *        WITH A FORMAT WHICH DEPENDS ON HOW THE TRANSFORMER
060 *        IS CONNECTED AS INDICATED BY THE PARAMETER IPHSE.
070 *        NOTE: ASSUMES CONSISTANCY OF IPHSE AND NUMF
080 *        CHECKED IN CALLING PROGRAM
090        COMPLEX Y,ZERO,YT(NDIM,NDIM)
100        DATA ZERO/(0.0,0.0)/
110        DO 10 I=1,NDIM
120        DO 10 J=1,NDIM
130        YT(I,J)=ZERO
140     10 CONTINUE
150        IPHSM=IPHSE
160        IF(IPHSM.EQ.0)IPHSM=1
170        IF(IPHSM.GT.3)GOTO20
180        YT(IPHSM,IPHSM)=Y
190        RETURN
200     20 J=IPHSM/2-1
210        K=MOD((IPHSM-J+1),3)+1
220 *        THE ABOVE LOGIC ESTABLISHES THE FOLLOWING
230 *          IPHSM  J  K
240 *            4    1  2
250 *            5    1  3
260 *            6    2  3
270 *            7    2  1
280 *            8    3  1
290 *            9    3  2
300        YT(J,J)=Y
310        YT(K,K)=Y
320        YT(J,K)=-Y
330        YT(K,J)=-Y
340        RETURN
350        END
```

```
**********************************************************************

YTRANPS2                                          04/08/81     4:01 PM

**********************************************************************


010**RUN *=$YTRANPO2(NOGO,CORE=30K)
020       SUBROUTINE YTRANP(YIN,PSVRL,QLOAD,ITT,LTRAN)
030 *       THIS SUBROUTINE CALCULATES THE SCALAR COMPLEX INPUT
040 *       ADMITTANCE, YIN, AND PRIMARY TO SECONDARY VOLTAGE
050 *       TRANSFER RATIO, PSVRL, FOR A SINGLE PHASE
060 *       DISTRIBUTION TRANSFORMER.  IT IS TO BE USED IN
070 *       CONJUNCTION WITH A RANDOM BINARY DATAFILE DATRANIJ
080 *       CONTAINING THE ALGEBRAIC MODEL PARAMETERS AS
090 *       DEVELOPED BY PROF. J.T.GAJJAR IN DEC.1979
100 *       FOR DETAILS SEE VEMO REPORT ON TRANSFORMER MODELS
110 *         YL    =TWICE LINE TO LINE SECONDARY LOAD ADMITTANCE
120 *         QLOAD=FRACTIONAL PERUNIT COMPLEX LOAD
130 *         Z11   =PRIMARY ADMITTANCE WITH SECODARY OPEN CIRCUIT
140 *         PSVRO=PRIMARY TO SECONDARY VOLTAGE RATIO OPEN CIRCUIT
150 *         PSVRL=PRIMARY TO SECONDARY VOLTAGE RATIO WITH LOAD
160 *         ITT   =TRANSFORMER IDENTIFICATION NUMBER
170 *         LTRAN=FRN FOR TRANSFORMER DATRANIJ FILE
180 *         ZA,ZB=NOT USED FOR NOW
190 *         IDUM =NOT USED FOR NOW
200       COMPLEX YIN,YL,Y11,Y12,Y22,Y23,Z11,ZA,ZB,PSVRO,PSVRL,QLOAD
210       READ(LTRAN'ITT+1)I,PKVA,IDUM,Y11,Y12,Y22,Y23,Z11,PSVRO,ZA,ZB
220       IF(I.NE.ITT)CALL FILSTO(MTRAN,'RECORD',ITT,I,10110)
230       YL=QLOAD*PKVA/28.8
240       PSVRL=PSVRO*Y12/(Y12-PSVRO*YL)
250       YIN=Z11+YL*(Y11-Z11)/(YL+Y22-Y23)
260 *       ALTERNATE FORMULATION
270 *     YIN=Z11-YL*(Y11-Z11)*PSVRL/Y12
280       RETURN
290       END
```

```
0010*#RUN *=$ZPLOTMC1(NOGO,CORE=30K)
0020       SUBROUTINE ZPLOTM(ID,NP,MV,A,NDIM,MDIM,IXC,IYC,ITI,X1,X2,
0030      &                                                NXC,NYC,NTI,Y1,Y2)
0040 *       THIS SUBROUTINE GENERATES A ZETA PLOT FILE WITH NAME RRPLTIJK
0050 *       WHERE IJK IS THREE DIGIT ID.  THIS FILE WILL PLOT M CURVES,
0060 *       CORRESPONDING TO THE COLUMNS 2 THRU M+1 OF REAL ARRAY, A,
0070 *       VS DEPENDANT VARAIBLE CONTAINED IN COLUMN 1 OF A
0080 *
0090 *         INDEPENDANT VARIABLE ASSUMED TO LIE IN RANGE (X1,X2)
0100 *           DEPENDANT VARIABLE ASSUMED TO LIE IN RANGE (Y1,Y2)
0110 *
0120 *       ID    CASE IDENTIFICATION NUMBER-THREE DIGITS
0130 *       MV    NUMBER OF DEPENDANT VARIABLES
0140 *       NP    NUMBER OF POINTS ON ABCISSA
0150 *       NDIM  ROW DIMENSION OF A
0160 *       MDIM  COLUMN DIMENSION OF A
0170 *       IXC   CHARACTER VARIABLE CONTAINING ABCISSA  LEGEND
0180 *       IYC   CHARACTER VARIABLE CONTAINING ORDINATE LEGEND
0190 *       ITI   CHARACTER VARIABLE CONTAINING TITLE
0200 *       NXC   NUMBER OF CHARACTERS IN XCV
0210 *       NYC   NUMBER OF CHARACTERS IN YCV
0220 *       NTI   NUMBER OF CHARACTERS IN ITI
0221 *       X1    ABCISSA LOWER LIMIT-PHYSICAL UNITS
0222 *       X2    ABCISSA UPPER LIMIT-PHYSICAL UNITS
0223 *       Y1    ORDINATE LOWER LIMIT-PHYSICAL UNITS
0224 *       Y2    ORDINATE UPPER LIMIT-PHYSICAL UNITS
0230 *
0240       DIMENSION A(NDIM,MDIM),IV(3)
0250       CHARACTER FILE*9,IXC*30,IYC*30,ITI*30,ICNO*3,NDATE*8
0260       DATA AXL/8.0/,AYL/6.0/,IV/2,4,5/,XBIG/1.E20/
0270    11 FORMAT('RRPLT',I3,'$')
0280    12 FORMAT(I3)
0281    13 FORMAT('WARNING IN ZPLOTN,',A2,' VALUES OUT OF RANGE')
0290       XPF(X)=(X-XO)/XF
0300       YPF(Y)=(Y-YO)/YF
0310       IF(MV.LT.MDIM.AND.NP.LE.NDIM)GOTO220
0320       PRINT,'STOPPED IN ZPLOTN, MV OR NP TOO LARGE'
0330       PRINT,'MV,MDIM,NP,NDIM=',MV,MDIM,NP,NDIM
0340       STOP 0220
0350   220 CONTINUE
0360       CALL DATIM(NDATE,TIME)
0370 *       CHECK X VALUES IN RANGE
0380       XMIN=XBIG
0390       XMAX=-XBIG
0400       IFL=0
0410       DO 230 I=1,NP
0420       X=A(I,1)
0430       IF(X.GE.X1.AND.X.LE.X2)GOTO230
0440       XMIN=AMIN1(XMIN,X)
```

```
0450        XMAX=AMAX1(XMAX,X)
0460        IFL=1
0470    230 CONTINUE
0480        IF(IFL.EQ.0)GOTO240
0490        PRINT 13,'X'
0500        PRINT,'XMIN=',XMIN,' XMAX=',XMAX
0510    240 CONTINUE
0520        XMIN=XBIG
0530        XMAX=-XBIG
0540        IFL=0
0550        DO 250 I=1,NP
0560        DO 250 J=2,MV+1
0570        X=A(I,J)
0580        IF(X.GE.Y1.AND.X.LE.Y2)GOTO250
0590        XMIN=AMIN1(XMIN,X)
0600        XMAX=AMAX1(XMAX,X)
0610        IFL=1
0620    250 CONTINUE
0630        IF(IFL.EQ.0)GOTO260
0640        PRINT 13,'Y'
0650        PRINT,'YMIN=',XMIN,'      YMAX=',XMAX
0660    260 CONTINUE
0670        CALL NASTRK
0680        ENCODE(FILEN,11)ID+1000
0690        ENCODE(ICNO,12)ID+1000
0700        CALL YASTRK
0710 *        AXL AND AYL PERTAIN TO THE ABCISSA AND ORDINATE RESPECTIVELY
0720 *        NOTE 90 DEGREE ROTATION ASSOCIATED WITH USE IN XF AND YF
0730        YF=(X2-X1)/AXL
0740        XF=(Y2-Y1)/AYL
0750        XO=-Y1-7.5*XF
0760        YO=X1-1.5*YF
0770 *        LETTING XP AND YP BE PAPER COORDINATES IN INCHES, THEN FOR
0780 *        90 DEGREE ROTATION
0790 *          YP=1.5+(X-X1)/YF=(X-X1+1.5*YF)/YF,  YO=X1-1.5*YF
0800 *          XP=7.5-(Y-Y1)/XF=(-Y+Y1+7.5*XF)/XF, XO=-Y1-7.5*XF
0810 *        NOTE NEGATIVE ENTRY ON ORDINATE VALUE
0820        CALL PLOTF(30,FILEN,ISTAT)
0830        IF(ISTAT.EQ.0)GOTO300
0840        PRINT,'PLOTF ERROR',ISTAT
0850        STOP 0300
0860    300 CONTINUE
0861        XNTI = (30.0-NTI)/2.0
0862        XNTI = XNTI*0.3 + 1.0
0870        CALL SYMBOL(1.0,XNTI,0.2,ITI,90.0,NTI)
0880        CALL SYMBOL(1.5,4.0,0.2,'CASENO=',90.0,7)
0890        CALL SYMBOL(1.5,6.2,0.2,ICNO,90.0,3)
0900        CALL SYMBOL(8.3,7.0,.1,NDATE,90.0,8)
0910        CALL NUMBER(8.3,6.5,.1,TIME,90.0,3)
0920        CALL AXIS(7.5,1.5,IXC,-NXC,AXL,90.0,X1,YF)
0930        CALL AXIS(7.5,1.5,IYC,NYC,AYL,180.0,Y1,XF)
0940 *        NOTE MINUS SIGN Y AXIS DUE TO 90 DEGREE AND OFFSET GRAMMAR
```

```
0950          CALL PLOTI(2.5)
0960          DO 310 J=2,MV+1
0970          IC=IV(MOD(J,3)+1)
0980          I=1
0990          XP=XPF(-A(I,J))
1000          YP=YPF(A(I,1))
1010          CALL PLOT(XP,YP,3)
1020          DO 310 I=2,MP
1030          XP=XPF(-A(I,J))
1040          YP=YPF(A(I,1))
1050      310 CALL PLOT(XP,YP,IC)
1060          CALL PLOT(8.5,0.0,-3)
1070          CALL PLOT(0.0,0.0,999)
1080          PRINT,'WROTE PLOT FILE ',FILEN
1090          RETURN
1100          END
```

C-3

```
**********************************************************************

ZPLOTTS1                                          04/08/81    4:01 PM

**********************************************************************


0010*#RUN *=:ZPLOTT01(NOGO,CORE=30K)
0020       SUBROUTINE ZPLOTT(ID,NP,MV,A,NDIM,MDIM,IXC,IYC,ITI,XO,
0030      &                  X1,X2,XS,IFO,NXC,NYC,NTI,YO,Y1,Y2,YD)
0040 *       THIS SUBROUTINE GENERATES A ZETA PLOT FILE WITH NAME RPLOTIJK
0050 *       WHERE IJK IS THREE DIGIT ID.  THIS FILE WILL PLOT M CURVES
0060 *       CORRESPONDING TO THE COLUMNS 2 THRU M+1 OF REAL ARRAY, A,
0070 *       VS DEPENDANT VARAIBLE CONTAINED IN COLUMN 1 OF A
0080 *
0090 *         INDEPENDENT VARIABLE ASSUMED TO LIE IN RANGE (X1,X2)
0100 *           DEPENDENT VARIABLE ASSUMED TO LIE IN RANGE (Y1,Y2)
0110 *
0120 *       THIS PLOT SUBROUTINE DIFFERS FROM ZPLOTM IN THAT THE
0130 *       THE ABCISSA AND ORDINATE DIRECTIONS RELATIVE TO THE
0140 *       GRAPH PAPER ARE ROTATED 90 DEGREES.  THUS, THE ABCISSA IS
0150 *       ESSENTIALLY UNLIMITED IN LENGTH BECAUSE IT RUNS IN THE
0160 *       DIRECTION OF THE PAPER FEED.
0170 *       OF SIGNIFICANT DIFFERENCE IS THE ADDITION OF THE ARG-
0180 *       UMENT 'IFO', WHICH ALLOWS THE USER MORE CONTROL OVER THE
0190 *       OUTPUT OF THE ZETA PLOTTER.  IFO = 0 SHOULD BE THE
0200 *       DEFAULT VALUE SINCE IT RESETS THE ORIGIN AND TURNS OFF
0210 *       THE PLOTTER AT THE END OF EXECUTION.  IFO = 1 ALLOWS THE
0220 *       USER TO PUT TWO LABELS ON THE ORDINATE AND PLOT TWO
0230 *       SEPERATE FUNCTIONS ALONG THE SAME ABCISSA.  IFO = 2 ALLOWS
0240 *       THE USER TO SET A NEW ORIGIN AND PLOT ANOTHER GRAPH IN
0250 *       THE SAME OUTPUT FILE.  IT MUST BE NOTED THAT ONLY IFO = 0
0260 *       TURNS OFF THE PLOTTER SO THE FINAL CALL TO THIS ROUTINE
0270 *       SHOULD BE WITH IFO = 0.
0280
0290 *       ID     CASE IDENTIFICATION
0300 *       MV     NUMBER OF DEPENDENT VARIABLES
0310 *       NP     NUMBER OF POINTS OF ABCISSA
0320 *       NDIM   ROW DIMENSION OF A
0330 *       MDIM   COLUMN DIMENSION OF A
0340 *       IXC    CHARACTER VARIABLE CONTAINING ABCISSA LEGEND
0350 *       IYC    CHARACTER VARIABLE CONTAINING ORDINATE LEGEND
0360 *       ITI    CHARACTER VARIABLE CONTAINING TITLE
0370 *       NXC    NUMBER OF CHARACTERS IN IXC
0380 *       NYC    NUMBER OF CHARACTERS IN IYC
0390 *       NTI    NUMBER OF CHARACTERS IN ITI
0400 *       IFO    =0  RESET ORIGIN AND TURN OFF PLOTTER BEFORE RETURNING
0410 *              =1  DO NOT RESET ORIGIN AND DO NOT TURN OFF PLOTTER
0420 *                  (USED FOR MULTIPLE PLOTS ON SAME GRAPH)
0430 *              =2  RESET ORIGIN BUT DO NOT TURN OFF PLOTTER
0440 *                  (USED FOR MULTIPLE GRAPHS IN SAME FILE)
0450 *       XO     ABCISSA COORDINATE OF ORIGIN
0460 *       YO     ORDINATE COORDINATE OF ORIGIN
0470 *       X1     ABCISSA LOWER LIMIT IN PHYSICAL UNITS
0480 *       X2     ABCISSA UPPER LIMIT IN PHYSICAL UNITS
0490 *       XS     ABCISSA SCALE UNITS/INCH
```

```
0500 *        Y1     ORDINATE LOWER LIMIT
0510 *        Y2     ORDINATE UPPER LIMIT
0520 *        YS     ORDINATE SCALE UNITS/INCH-DETERMINED IN THIS SUBROUTINE
0530 *               BASED ON INPUT YD
0540 *        YD     ORDINATE DISTANCE IN INCHES TO BE SPANNED BY Y2-Y1
0550     11 FORMAT('RPLOT',I3,';')
0560     12 FORMAT(I3)
0570     13 FORMAT('WARNING IN ZPLOTT,',A2,' VALUES ARE OUT OF RANGE')
0580        DIMENSION A(MDIM,MDIM),IV(3)
0590        CHARACTER FILE*9,IXC*30,IYC*30,ITI*30,ICNO*3,NDATE*8
0600        INTEGER ONFLAG/0/,LABELED/0/,LABEL2/0/,LASTYPE/0/
0610        DATA XBIG/1.E20/,IV/2,4,5/,JCOUNT/1/,IB/0/
0620        XPF(X)=XO+(X-X1)/XS
0630        YPF(Y)=YO+(Y-Y1)/YS
0640        YS=(Y2-Y1)/YD
0650        IF(MV.LT.MDIM.AND.NP.LE.MDIM)GOTO220
0660        PRINT,'STOPPED IN ZPLOTT, MV OR NP TOO LARGE'
0670        PRINT,'MV,MDIM,NP,MDIM=',MV,MDIM,NP,MDIM
0680        STOP 0220
0690    220 CONTINUE
0700        CALL DATIM(NDATE,TIME)
0710        XMIN=XBIG
0720        XMAX=-XBIG
0730        IFL=0
0740        DO 230 I=1,NP
0750        X=A(I,1)
0760        IF(X.GE.X1.AND.X.LE.X2)GOTO230
0770        XMIN=AMIN1(XMIN,X)
0780        XMAX=AMAX1(XMAX,X)
0790        IFL=1
0800    230 CONTINUE
0810        IF(IFL.EQ.0)GOTO240
0820        PRINT 13,'X'
0830        PRINT,'XMIN=',XMIN,' XMAX=',XMAX
0840    240 CONTINUE
0850        XMIN=XBIG
0860        XMAX=-XBIG
0870        IFL=0
0880        DO 250 I=1,NP
0890        DO 250 J=2,MV+1
0900        X=A(I,J)
0910        IF(X.GE.Y1.AND.X.LE.Y2)GOTO250
0920        XMIN=AMIN1(XMIN,X)
0930        XMAX=AMAX1(XMAX,X)
0940        IFLG=1
0950    250 CONTINUE
0960        IF(IFLG.EQ.0)GOTO260
0970        PRINT 13,'Y'
0980        PRINT,'YMIN=',XMIN,'YMAX=',XMAX
0990        IFLG=0
1000    260 CONTINUE
1010        CALL NASTRK
```

```
1020        ENCODE(FILEN,11)ID+1000
1030        ENCODE(ICNO,12)ID+1000
1040        CALL YASTRK
1050
1060  *           TURN ON PLOTTER ONLY IF THIS IS THE FIRST
1070  *           GRAPH IN A NEW FILE.
1080
1090        IF(ONFLAG.EQ.1)GOTO265
1100        CALL PLOTF(30,FILEN,ISTAT)
1110        IF(ISTAT.EQ.0)GOTO265
1120        PRINT,'PLOTF ERROR',ISTAT
1130        STOP 0265
1140    265 CONTINUE
1150        CALL PLOTI(10.0)
1160        ONFLAG=1
1170        IF(LABELED.EQ.1)GOTO267
1180        CALL SYMBOL(2.0,10.0,0.2,ITI,0.0,NTI)
1190        CALL SYMBOL(3.5,9.5,0.2,'CASENO=',0.0,7)
1200        CALL SYMBOL(5.7,9.5,0.2,ICNO,0.0,3)
1210        CALL SYMBOL(14.0,0.1,.1,IDATE,0.0,8)
1220        CALL NUMBER(15.5,0.1,.1,TIME,0.0,3)
1230        CALL AXIS(XO,YO,IYC,-NYC,(Y2-Y1)/YS,90.0,Y1,YS)
1240        CALL AXIS(XO,YO,IXC,-NXC,(X2-X1)/XS,0.0,X1,XS)
1250        LABELED=1
1260    267 IF(LASTYPE.NE.1.OR.LABEL2.EQ.1)GOTO270
1270        CALL AXIS(XO,YO,IYC,-NYC,(Y2-Y1)/YS,90.0,Y1,YS)
1280        LABEL2=1
1290    270 CONTINUE
1300        CALL PLOTI(2.5)
1310        DO 315 J=2,NV+1
1320        IC=IV(MOD(JCOUNT,3)+1 )
1330        I=1
1340        XP=XPF(A(I,1))
1350        YP=YPF(A(I,J))
1360        CALL PLOT(XP,YP,3)
1370        DO 310 I=2,NP
1380        XP=XPF(A(I,1))
1390        YP=YPF(A(I,J))
1400        CALL PLOT(XP,YP,IC)
1410    310 CONTINUE
1420        JCOUNT=JCOUNT+1
1430    315 CONTINUE
1440  *       NOW PERFORM CLOSING AS INDICATED BY THE USER'S
1450  *       SPECIFICATION OF 'IFO'.
1460        IF(IFO.GE.0.AND.IFO.LE.2)GOTO320
1470        PRINT,'ERROR IN ARGUMENT IFO',IFO
1480        STOP 0320
1490
1500    320 CONTINUE
1510        IF(IFO.EQ.1)GOTO340
1520        IF(IFO.EQ.2)GOTO330
1530        ONFLAG=0
```

3. ILLUSTRATIVE ASCII DATA BASE FILES

```
**********************************************************************

AFPRY886                                          04/16/81    2:56 PM

**********************************************************************

 1000  IFRE=88  ITMP=6  LREC=38  IPRM= 55  NDATE= 04/16/81  TIME= 14.90
 1010
 1020   41     3     0            3     0     0
 1030
 1040     4.4100000E-05      1.4300000E-05      1.4300000E-05
 1050    -6.2000000E-04      1.0200000E-05      1.0200000E-05
 1060
 1070     1.4300000E-05      4.4100000E-05      1.4300000E-05
 1080     1.0200000E-05     -6.2000000E-04      1.0200000E-05
 1090
 1100     1.4300000E-05      1.4300000E-05      4.4100000E-05
 1110     1.0200000E-05      1.0200000E-05     -6.2000000E-04
 1120
 1130   42     3     0            3     0     0
 1140
 1150     1.8800000E-04     -1.2300000E-05     -1.2300000E-05
 1160    -2.9700000E-05      6.8900000E-05      6.8900000E-05
 1170
 1180    -1.2300000E-05      1.8800000E-04     -1.2300000E-05
 1190     6.8900000E-05     -2.9700000E-05      6.8900000E-05
 1200
 1210    -1.2300000E-05     -1.2300000E-05      1.8800000E-04
 1220     6.8900000E-05      6.8900000E-05     -2.9700000E-05
 1230
 1240   43     3     0            3     0     0
 1250
 1260     8.5800000E-05     -8.4600000E-07     -8.4600000E-07
 1270    -1.6100000E-04      3.3000000E-05      3.3000000E-05
 1280
 1290    -8.4600000E-07      8.5800000E-05     -8.4600000E-07
 1300     3.3000000E-05     -1.6100000E-04      3.3000000E-05
 1310
 1320    -8.4600000E-07     -8.4600000E-07      8.5800000E-05
 1330     3.3000000E-05      3.3000000E-05     -1.6100000E-04
 1340
 1350   44     3     0            3     0     0
 1360
 1370     8.1100000E-05     -1.1400000E-05     -1.1400000E-05
 1380    -9.8600000E-05      2.6000000E-05      2.6000000E-05
 1390
 1400    -1.1400000E-05      8.1100000E-05     -1.1400000E-05
 1410     2.6000000E-05     -9.8600000E-05      2.6000000E-05
 1420
 1430    -1.1400000E-05     -1.1400000E-05      8.1100000E-05
 1440     2.6000000E-05      2.6000000E-05     -9.8600000E-05
 1450
 1460   45     3     0            3     0     0
 1470
 1480     6.4000000E-05     -9.5100000E-06     -9.5100000E-06
```

This file has been obtained by reformatting file R2 (after first renaming R2 to AFPRY886) using main program AFPRYGEN.

```
1490    -7.5500000E-05      2.0100000E-05      2.0100000E-05
1500
1510    -9.5100000E-06      6.4000000E-05     -9.5100000E-06
1520     2.0100000E-05     -7.5500000E-05      2.0100000E-05
1530
1540    -9.5100000E-06     -9.5100000E-06      6.4000000E-05
1550     2.0100000E-05      2.0100000E-05     -7.5500000E-05
1560
1570    46    3    0           3    0    0
1580
1590     4.8400000E-0       -8.5200001E-06     -8.5200001E-06
1600    -5.0600000E-05      1.4000000E-05      1.4000000E-05
1610
1620    -8.5200001E-06      4.8400000E-05     -8.5200001E-06
1630     1.4000000E-05     -5.0600000E-05      1.4000000E-05
1640
1650    -8.5200001E-06     -8.5200001E-06      4.8400000E-05
1660     1.4000000E-05      1.4000000E-05     -5.0600000E-05
1670
1680    47    3    0           3    0    0
1690
1700     3.3800000E-05     -6.9200000E-06     -6.9200000E-06
1710    -3.0400000E-05      8.6000000E-06      8.6000000E-06
1720
1730    -6.9200000E-06      3.3800000E-05     -6.9200000E-06
1740     8.6000000E-06     -3.0400000E-05      8.6000000E-06
1750
1760    -6.9200000E-06     -6.9200000E-06      3.3800000E-05
1770     8.6000000E-06      8.6000000E-06     -3.0400000E-05
1780
1790    48    3    0           3    0    0
1800
1810     2.0100000E-05     -3.6800000E-06     -3.6800000E-06
1820    -2.0300000E-05      5.6400000E-06      5.6400000E-06
1830
1840    -3.6800000E-06      2.0100000E-05     -3.6800000E-06
1850     5.6400000E-06     -2.0300000E-05      5.6400000E-06
1860
1870    -3.6800000E-06     -3.6800000E-06      2.0100000E-05
1880     5.6400000E-06      5.6400000E-06     -2.0300000E-05
1890
1900    49    3    0           3    0    0
1910
1920     2.4200000E-05      2.4200000E-05      2.4200000E-05
1930     1.5500000E-01     -7.8400000E-02     -7.8400000E-02
1940
1950     2.4200000E-05      2.4200000E-05      2.4200000E-05
1960    -7.8400000E-02      1.5500000E-01     -7.8400000E-02
1970
1980     2.4200000E-05      2.4200000E-05      2.4200000E-05
1990    -7.8400000E-02     -7.8400000E-02      1.5500000E-01
2000
```

```
2010    51     3     0          3     0     0
2020
2030      2.4600000E-05     2.4600000E-05     2.4600000E-05
2040      7.7100000E-02    -3.9400000E-02    -3.9400000E-02
2050
2060      2.4600000E-05     2.4600000E-05     2.4600000E-05
2070     -3.9400000E-02     7.7100000E-02    -3.9400000E-02
2080
2090      2.4600000E-05     2.4600000E-05     2.4600000E-05
2100     -3.9400000E-02    -3.9400000E-02     7.7100000E-02
2110
2120    55     3     0          3     0     0
2130
2140      2.4100000E-05     2.4100000E-05     2.4100000E-05
2150      3.1100000E-01    -1.5600000E-01    -1.5600000E-01
2160
2170      2.4100000E-05     2.4100000E-05     2.4100000E-05
2180     -1.5600000E-01     3.1100000E-01    -1.5600000E-01
2190
2200      2.4100000E-05     2.4100000E-05     2.4100000E-05
2210     -1.5600000E-01    -1.5600000E-01     3.1100000E-01
```

1000	1												
1001	1	0	2	0	101	2000	4	25	1	0	0	50	2000
1002	2	1	3	0	101	2000	4	25	2	0	0	50	4000
1003	3	2	4	0	101	2000	4	25	3	0	0	50	6000
1004	4	3	5	0	101	2000	4	25	1	0	0	50	8000
1005	5	4	6	0	101	2000	4	25	2	0	0	50	10000
1006	6	5	7	0	101	2000	4	25	3	0	0	50	12000
1007	7	6	8	0	101	2000	4	25	1	0	0	50	14000
1008	8	7	9	0	101	2000	4	25	2	0	0	50	16000
1009	9	8	10	0	101	2000	4	25	3	0	0	50	18000
1010	10	9	11	0	101	2000	4	25	1	0	0	50	20000
1011	11	10	12	0	101	2000	4	25	2	0	0	50	22000
1012	12	11	13	0	101	2000	4	25	3	0	0	50	24000
1013	13	12	14	0	101	2000	4	25	1	0	0	50	26000
1014	14	13	15	0	101	2000	4	25	2	0	0	50	28000
1015	15	14	16	0	101	2000	4	25	3	0	0	50	30000
1016	16	15	17	0	101	2000	4	25	1	0	0	50	32000
1017	17	16	18	0	101	2000	4	25	2	0	0	50	34000
1018	18	17	19	0	101	2000	4	25	3	0	0	50	36000
1019	19	18	20	0	101	2000	4	25	1	0	0	50	38000
1020	20	19	21	0	101	2000	4	25	2	0	0	50	40000
1021	21	20	22	0	101	2000	4	25	3	0	0	50	42000
1022	22	21	23	0	101	2000	4	25	1	0	0	50	44000
1023	23	22	24	0	101	2000	4	25	2	0	0	50	46000
1024	24	23	25	0	101	2000	4	25	3	0	0	50	48000
1025	25	24	26	0	101	2000	4	25	1	0	0	50	50000
1026	26	25	27	0	101	2000	4	25	2	0	0	50	52000
1027	27	26	28	0	101	2000	4	25	3	0	0	50	54000
1028	28	27	29	0	101	2000	4	25	1	0	0	50	56000
1029	29	28	30	0	101	2000	4	25	2	0	0	50	58000
1030	30	29	31	0	101	2000	4	25	3	0	0	50	60000
1031	31	30	32	0	101	2000	4	25	1	0	0	50	62000
1032	32	31	33	0	101	2000	4	25	2	0	0	50	64000
1033	33	32	34	0	101	2000	4	25	3	0	0	50	66000
1034	34	33	35	0	101	2000	4	25	1	0	0	50	68000
1035	35	34	36	0	101	2000	4	25	2	0	0	50	70000
1036	36	35	37	0	101	2000	4	25	3	0	0	50	72000
1037	37	36	38	0	101	2000	4	25	1	0	0	50	74000
1038	38	37	39	0	101	2000	4	25	2	0	0	50	76000
1039	39	38	40	0	101	2000	4	25	3	0	0	50	78000
1040	40	39	41	0	101	2000	4	25	1	0	0	50	80000
1041	41	40	42	0	101	2000	4	25	2	0	0	50	82000
1042	42	41	43	0	101	2000	4	25	3	0	0	50	84000
1043	43	42	44	0	101	2000	4	25	1	0	0	50	86000
1044	44	43	45	0	101	2000	4	25	2	0	0	50	88000
1045	45	44	46	0	101	2000	4	25	3	0	0	50	90000
1046	46	45	47	0	101	2000	4	25	1	0	0	50	92000
1047	47	46	48	0	101	2000	4	25	2	0	0	50	94000
1048	48	47	49	0	101	2000	4	25	3	0	0	50	96000

1049	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	1	0	0	50	98000
1050	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	2	0	0	50	100000
1051	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	3	0	0	50	102000
1052	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	1	0	0	50	104000
1053	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	2	0	0	50	106000
1054	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	3	0	0	50	108000
1055	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	1	0	0	50	110000
1056	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	2	0	0	50	112000
1057	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	3	0	0	50	114000
1058	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	1	0	0	50	116000
1059	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	2	0	0	50	118000
1060	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	3	0	0	50	120000
1061	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	1	0	0	50	122000
1062	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	2	0	0	50	124000
1063	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	3	0	0	50	126000
1064	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	1	0	0	50	128000
1065	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	2	0	0	50	130000
1066	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	3	0	0	50	132000
1067	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	1	0	0	50	134000
1068	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	2	0	0	50	136000
1069	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	3	0	0	50	138000
1070	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	1	0	0	50	140000
1071	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	2	0	0	50	142000
1072	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	3	0	0	50	144000
1073	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	1	0	0	50	146000
1074	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	2	0	0	50	148000
1075	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	3	0	0	50	150000
1076	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	1	0	0	50	152000
1077	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	2	0	0	50	154000
1078	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	3	0	0	50	156000
1079	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	1	0	0	50	158000
1080	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	4	25	2	0	1	50	160000

****

DNWKIN72 09/07/81 10:25 AM

****

1000	89											
1001	1	946	2	0	101	1056	4	10	3	0	0	0
1002	2	1	3	0	101	4252	4	25	3	0	0	0
1003	3	2	4	0	101	2185	4	0	0	0	0	0
1004	4	3	5	0	103	646	4	25	3	0	0	0
1005	5	4	6	0	109	1490	4	10	2	0	0	0
1006	6	5	7	0	109	838	4	0	0	0	0	0
1007	7	6	8	0	107	1103	4	15	1	0	0	0
1008	8	7	9	0	107	1646	4	0	0	0	0	0
1009	9	8	1308	0	109	1231	4	15	2	0	0	0
1010	1308	9	1309	0	109	2109	4	15	1	0	0	0
1011	1309	1308	10	0	109	672	4	10	1	0	0	0
1012	10	1309	11	0	105	3360	4	10	2	0	0	0
1013	11	10	12	14	105	896	4	0	0	0	0	0
1014	12	11	13	0	1	0	4	0	0	0	0	0
1015	13	12	0	0	463	1350	4	15	1	0	0	0
1016	14	11	15	0	105	762	4	0	0	0	0	0
1017	15	14	16	0	107	802	4	0	0	0	0	0
1018	16	15	17	0	105	2423	4	25	1	0	0	0
1019	17	16	18	30	105	768	4	0	0	0	0	0
1020	18	17	0	20	2	0	4	0	2	0	0	0
1021	20	18	21	0	313	1104	4	10	1	0	0	0
1022	21	20	22	0	313	1440	4	15	1	0	0	0
1023	22	21	23	0	313	1056	4	25	1	0	0	0
1024	23	22	24	0	313	1248	4	15	1	0	0	0
1025	24	23	25	0	313	1530	4	10	1	0	0	0
1026	25	24	0	0	313	1632	4	15	1	0	0	0
1027	30	17	31	0	101	1497	4	10	1	0	0	0
1028	31	30	32	0	103	1500	4	0	0	0	0	0
1029	32	31	33	0	101	1920	4	10	3	0	0	0
1030	33	32	34	0	109	1012	4	0	0	0	0	0
1031	34	33	35	0	109	1917	4	10	3	0	0	0
1032	35	34	36	0	109	1530	4	0	0	0	0	0
1033	36	35	37	0	155	1012	4	10	2	0	0	0
1034	37	36	38	0	157	570	4	0	0	0	0	0
1035	38	37	39	0	155	1308	4	5	1	0	0	0
1036	39	38	40	0	155	1824	4	0	0	0	0	0
1037	40	39	41	250	121	480	4	0	0	0	0	0
1038	41	40	42	0	101	0	4	0	0	0	0	0
1039	42	41	45	43	121	0	4	0	0	0	0	0
1040	43	42	44	0	101	1248	4	15	2	0	0	0
1041	44	43	100	0	101	610	4	0	0	0	0	0
1042	100	44	101	102	601	1445	4	0	0	0	0	0
1043	101	100	1300	0	525	5000	4	0	0	0	0	0
1044	102	100	103	0	213	799	4	0	0	0	0	0
1045	103	102	104	0	3	0	4	0	0	0	0	0
1046	104	103	105	0	313	960	4	5	1	0	0	0
1047	105	104	106	0	313	1750	4	15	1	0	0	0
1048	106	105	107	0	313	1862	4	5	1	0	0	0

1049	107	106	108	0	313	2062	4	5	1	0	0	0
1050	108	107	109	0	303	872	4	5	1	0	0	0
1051	109	108	110	0	313	1824	4	0	0	0	0	0
1052	110	109	111	0	315	536	4	0	0	0	0	0
1053	111	110	112	0	313	1521	4	10	1	0	0	0
1054	112	111	113	0	315	527	4	0	0	0	0	0
1055	113	112	114	0	313	939	4	15	1	0	0	0
1056	114	113	115	0	313	1210	4	25	1	0	0	0
1057	115	114	116	0	313	2890	4	10	1	0	0	0
1058	116	115	0	0	313	908	4	15	1	0	0	0
1059	45	42	46	0	117	1564	4	0	0	0	0	0
1060	46	45	0	0	101	5576	4	0	0	0	0	0
1061	250	40	251	0	119	3260	4	10	3	0	0	0
1062	251	250	253	249	119	1719	4	25	2	0	0	0
1063	249	251	252	0	3	0	4	0	0	0	0	0
1064	252	249	0	0	313	1152	4	25	1	0	0	0
1065	253	251	254	257	119	756	4	0	0	0	0	0
1066	254	253	255	0	1	0	4	0	1	0	0	0
1067	255	254	256	0	451	396	4	0	0	0	0	0
1068	256	255	0	0	453	1445	4	25	1	0	0	0
1069	257	253	258	0	115	1000	4	0	0	0	0	0
1070	258	257	259	0	119	1286	4	15	3	0	0	0
1071	259	258	260	0	119	2736	4	0	0	0	0	0
1072	260	259	261	264	118	960	4	25	1	0	0	0
1073	261	260	0	0	127	1765	4	0	0	0	43	0
1074	264	260	265	0	121	296	4	0	0	0	0	0
1075	265	264	274	280	109	561	4	25	2	0	0	0
1076	274	265	0	0	109	506	4	10	0	0	0	0
1077	280	265	283	281	109	2018	4	0	0	0	0	0
1078	281	280	282	0	3	0	4	0	0	0	0	0
1079	282	281	0	0	451	1440	4	40	1	0	0	0
1080	283	280	284	0	109	1164	4	0	0	0	49	0
1081	284	283	285	288	109	1747	4	0	0	0	0	0
1082	285	284	0	0	129	671	4	0	0	0	46	0
1083	288	284	289	0	111	960	4	15	2	0	0	0
1084	289	288	290	0	109	399	4	0	0	0	0	0
1085	290	289	291	0	109	4657	4	10	3	0	0	0
1086	291	290	292	0	109	1824	4	10	2	0	0	0
1087	292	291	293	0	109	672	4	40	2	0	0	0
1088	293	292	294	0	109	1056	4	10	2	0	0	0
1089	294	293	295	297	109	817	4	0	0	0	0	0
1090	295	294	296	0	1	0	4	0	0	0	0	0
1091	296	295	0	0	251	564	4	50	1	0	0	0
1092	297	294	298	0	109	811	4	0	0	0	0	0
1093	298	297	299	420	111	607	4	0	0	0	0	0
1094	299	298	300	0	109	1216	4	0	0	0	0	0
1095	300	299	301	0	111	1082	4	10	3	0	0	0
1096	301	300	302	0	109	472	4	0	0	0	0	0
1097	302	301	303	0	109	472	4	10	2	0	0	0
1098	303	302	304	0	109	604	4	25	2	0	0	0
1099	304	303	305	0	109	597	4	0	0	0	49	0
1100	305	304	306	0	109	597	4	0	0	0	0	0

1101	306	305	307	470	159	1186	4	15	2	0	0	0
1102	307	306	308	0	101	771	4	0	0	0	0	0
1103	308	307	309	0	102	394	4	5	3	0	0	0
1104	309	308	310	0	101	369	4	0	0	0	0	0
1105	310	309	311	0	110	1143	4	15	2	0	0	0
1106	311	310	312	0	109	314	4	0	0	0	0	0
1107	312	311	490	313	109	360	4	0	0	0	0	0
1108	313	312	316	0	109	564	4	0	0	0	47	0
1109	316	313	320	0	153	1070	4	25	1	0	0	0
1110	317	320	0	0	101	1119	4	0	0	0	47	0
1111	320	316	322	317	153	1140	4	0	0	0	0	0
1112	321	320	323	0	153	1248	4	25	1	0	0	0
1113	323	322	324	0	153	427	4	0	0	0	0	0
1114	324	323	601	325	151	2149	4	0	0	0	0	0
1115	325	324	326	0	153	585	4	10	1	0	0	0
1116	326	325	327	0	151	1253	4	15	3	0	0	0
1117	327	326	328	0	153	591	4	0	0	0	0	0
1118	328	327	329	331	151	1158	4	0	0	0	0	0
1119	329	328	330	0	1	0	4	0	0	0	0	0
1120	330	329	0	0	463	1113	4	5	1	0	0	0
1121	331	328	332	0	151	1661	4	10	2	0	0	0
1122	332	331	336	333	151	1414	4	0	0	0	0	0
1123	333	332	334	0	1	0	4	0	0	0	0	0
1124	334	333	335	0	459	850	4	0	0	0	0	0
1125	335	334	0	0	323	762	4	10	1	0	0	0
1126	336	332	337	0	153	384	4	25	1	0	0	0
1127	337	336	338	0	153	1265	4	25	3	0	0	0
1128	338	337	339	0	153	997	4	25	2	0	0	0
1129	339	338	340	0	153	1584	4	25	2	0	0	0
1130	340	339	341	0	153	1296	4	25	1	0	0	0
1131	341	340	342	0	153	451	4	0	0	0	0	0
1132	342	341	343	0	153	442	4	25	1	0	0	0
1133	343	342	344	0	153	439	4	25	2	0	0	0
1134	344	343	345	430	157	1157	4	0	0	0	0	0
1135	345	344	510	346	157	0	4	0	0	0	0	0
1136	346	345	347	0	157	1065	4	0	0	0	0	0
1137	347	346	348	0	161	744	4	15	3	0	0	0
1138	348	347	349	0	159	655	4	0	0	0	0	0
1139	349	348	350	361	159	3420	4	0	0	0	0	0
1140	351	350	1655	352	213	1056	4	0	0	0	0	0
1141	352	351	353	0	113	624	4	40	2	0	0	0
1142	353	352	354	0	101	1177	4	0	0	0	0	0
1143	354	353	355	0	213	1442	4	0	0	0	0	0
1144	355	354	356	0	215	439	4	15	2	0	0	0
1145	356	355	357	0	213	1789	4	15	2	0	0	0
1146	357	356	358	1015	213	672	4	15	2	0	0	0
1147	358	357	359	0	213	960	4	15	2	0	0	0
1148	359	358	360	0	213	360	4	15	2	0	0	0
1149	360	359	0	0	213	1050	4	15	3	0	0	0
1150	350	349	351	0	501	1968	4	0	0	0	0	0
1151	361	349	362	0	101	457	4	0	0	0	0	0
1152	362	361	363	0	603	430	4	10	3	0	0	0

1153	363	362	364	0	603	1067	4	0	0	0	0	0
1154	364	363	365	0	603	610	4	10	1	0	0	0
1155	365	364	366	0	503	646	4	0	0	0	0	0
1156	366	365	367	570	603	686	4	0	0	0	0	0
1157	367	366	368	0	603	686	4	15	1	0	0	0
1158	368	367	369	0	607	610	4	0	0	0	0	0
1159	369	368	370	0	607	619	4	10	2	0	0	0
1160	370	369	371	0	607	384	4	0	0	0	0	0
1161	371	370	1335	0	2	0	4	0	0	0	0	0
1162	1335	371	372	0	455	0	4	0	0	0	0	0
1163	372	1335	373	0	33	0	4	0	0	0	0	0
1164	420	298	421	0	105	658	4	0	0	0	0	0
1165	421	420	422	423	529	927	4	0	0	0	0	0
1166	422	421	1760	0	3	0	4	0	0	0	0	0
1167	423	421	424	0	125	610	4	25	2	0	0	0
1168	424	423	425	0	125	634	4	0	0	0	0	0
1169	425	424	426	0	129	920	4	10	3	0	0	0
1170	426	425	427	0	129	786	4	15	3	0	0	0
1171	427	426	428	457	129	1423	4	0	0	0	0	0
1172	428	427	429	450	118	2016	4	25	2	0	0	0
1173	429	428	430	0	129	552	4	0	0	0	0	0
1174	430	429	431	0	118	1448	4	40	1	0	0	0
1175	431	430	432	439	129	1728	4	0	0	0	0	0
1176	432	431	433	0	127	576	4	25	2	0	0	0
1177	433	432	434	1710	127	634	4	0	0	0	0	0
1178	434	433	435	1770	129	570	4	0	0	0	0	0
1179	435	434	436	0	129	564	4	15	1	0	0	0
1180	436	435	0	0	101	225	4	0	0	0	48	0
1181	439	431	440	0	2	0	4	0	0	0	0	0
1182	440	439	441	0	459	610	4	10	1	0	0	0
1183	441	440	442	0	459	707	4	25	1	0	0	0
1184	442	441	443	0	461	716	4	0	0	0	0	0
1185	443	442	444	0	453	610	4	25	1	0	0	0
1186	444	443	445	447	453	503	4	0	0	0	0	0
1187	445	444	446	0	451	518	4	0	0	0	0	0
1188	446	445	0	0	453	732	4	25	1	0	0	0
1189	447	444	448	0	453	1347	4	25	1	0	0	0
1190	448	447	449	0	453	732	4	25	1	0	0	0
1191	449	448	0	0	453	427	4	40	1	0	0	0
1192	450	428	451	0	3	0	4	0	0	0	0	0
1193	451	450	452	0	455	567	4	0	0	0	0	0
1194	452	451	453	0	457	610	4	25	1	0	0	0
1195	453	452	454	455	457	396	4	0	0	0	0	0
1196	454	453	0	0	459	564	4	25	1	0	0	0
1197	455	453	456	0	457	610	4	0	0	0	0	0
1198	456	455	0	0	457	732	4	25	1	0	0	0
1199	457	427	458	0	1	0	4	0	0	0	0	0
1200	458	457	459	0	459	604	4	40	1	0	0	0
1201	459	458	460	461	459	1440	4	0	0	0	0	0
1202	460	459	0	0	459	655	4	15	1	0	0	0
1203	461	459	462	0	459	671	4	15	1	0	0	0
1204	462	461	0	0	459	683	4	0	0	0	0	0

1205	470	306	471	0	3	0	4	0	0	0	0	0
1206	471	470	472	0	459	357	4	0	0	0	0	0
1207	472	471	473	0	459	433	4	25	1	0	0	0
1208	473	472	474	0	460	881	4	0	0	0	0	0
1209	474	473	475	0	459	442	4	25	1	0	0	0
1210	475	474	476	0	461	1152	4	15	1	0	0	0
1211	476	475	477	0	461	1021	4	25	1	0	0	0
1212	477	476	478	0	471	2356	4	25	1	0	0	0
1213	478	477	479	0	321	672	4	25	1	0	0	0
1214	1500	479	1501	0	905	686	4	50	1	0	0	0
12 5	1501	1500	1502	0	905	835	4	50	1	0	0	0
1216	1502	1501	1503	0	905	790	4	50	1	0	0	0
1217	1503	1502	1504	0	905	674	4	50	1	0	0	0
1218	1504	1503	1505	0	905	595	4	50	1	0	0	0
1219	1505	1504	1506	0	905	759	4	50	1	0	0	0
1220	1506	1505	1507	0	905	677	4	50	0	0	0	0
1221	1507	1506	0	0	905	999	4	50	0	0	0	0
1222	479	478	480	1500	313	2400	4	5	1	0	0	0
1223	480	479	481	0	313	864	4	10	1	0	0	0
1224	481	480	482	0	313	1481	4	5	1	0	0	0
1225	482	481	483	0	313	1644	4	0	0	0	0	0
1226	483	482	484	0	323	1006	4	5	1	0	0	0
1227	484	483	485	0	323	2241	4	5	1	0	0	0
1228	485	484	486	0	317	2208	4	15	1	0	0	0
1229	486	485	487	0	317	1152	4	15	1	0	0	0
1230	487	486	0	0	317	1632	4	5	1	0	0	0
1231	490	312	491	0	107	768	4	10	2	0	0	0
1232	491	490	0	492	605	564	4	0	0	0	0	0
1233	492	491	495	0	101	564	4	0	0	0	45	0
1234	495	492	498	0	505	762	4	0	0	0	47	0
1235	498	495	499	0	151	777	4	0	0	0	47	0
1236	499	498	500	0	151	1256	4	15	2	0	0	0
1237	500	499	0	0	101	539	4	0	0	0	0	0
1238	501	324	502	0	101	0	4	0	0	0	0	0
1239	502	501	503	0	101	1466	4	0	0	0	0	0
1240	503	502	504	0	101	1783	4	5	1	0	0	0
1241	504	503	505	1170	101	1509	4	0	0	0	0	0
1242	1170	504	1171	0	3	0	4	0	0	0	0	0
1243	1171	1170	0	0	905	700	4	0	0	0	0	0
1244	505	504	506	0	101	1183	4	15	1	0	0	0
1245	506	505	1720	0	101	779	4	0	0	0	0	0
1246	510	345	511	0	151	747	4	0	0	0	0	0
1247	511	510	517	1334	153	796	4	25	2	0	0	0
1248	1334	511	512	0	153	0	4	0	0	0	0	0
1249	512	1334	513	0	2	0	4	0	0	0	0	0
1250	513	512	514	0	455	64	4	0	0	0	0	0
1251	514	513	515	0	456	650	4	25	1	0	0	0
1252	515	514	516	0	456	1920	4	15	1	0	0	0
1253	516	515	0	0	457	442	4	0	0	0	0	0
1254	517	511	518	0	153	829	4	0	0	0	0	0
1255	518	517	519	0	153	582	4	0	0	0	0	0
1256	519	518	1336	0	1	0	4	0	0	0	0	0

1257	1336	519	1333	0	455	0	4	0	0	0	0	0
1258	1333	1336	520	0	33	0	4	0	0	0	0	0
1259	520	1333	521	0	401	305	4	40	4	0	0	0
1260	520	1333	521	0	403	305	4	0	0	0	0	0
1261	571	570	572	0	603	750	4	25	3	0	0	0
1262	572	571	573	0	503	997	4	25	1	0	0	0
1263	573	572	574	0	503	610	4	0	0	0	0	0
1264	574	573	575	585	503	579	4	0	0	0	0	0
1265	575	574	576	0	503	1237	4	25	3	0	0	0
1266	576	575	577	578	503	808	4	0	0	0	0	0
1267	577	576	578	0	2	0	4	0	0	0	0	0
1268	578	577	0	0	451	808	4	10	1	0	0	0
1269	579	576	580	0	503	1545	4	25	1	0	0	0
1270	580	579	581	583	503	792	4	0	0	0	0	0
1271	581	580	582	0	2	0	4	0	0	0	0	0
1272	582	581	0	0	451	808	4	15	1	0	0	0
1273	583	580	584	0	503	396	4	10	3	0	0	0
1274	584	583	0	0	503	672	4	0	0	0	0	0
1275	585	574	586	0	503	732	4	25	3	0	0	0
1276	586	585	587	0	103	570	4	0	0	0	0	0
1277	587	586	588	599	503	1122	4	0	0	0	0	0
1278	588	587	589	0	1	0	4	0	0	0	0	0
1279	589	588	590	0	453	570	4	25	1	0	0	0
1280	590	589	591	592	453	732	4	0	0	0	0	0
1281	591	590	0	0	451	945	4	10	1	0	0	0
1282	592	590	593	0	453	555	4	15	1	0	0	0
1283	593	592	594	595	453	549	4	0	0	0	0	0
1284	594	593	0	0	451	823	4	10	1	0	0	0
1285	595	593	596	0	451	792	4	0	0	0	0	0
1286	596	595	597	0	451	792	4	25	1	0	0	0
1287	597	596	598	0	451	640	4	15	1	0	0	0
1288	598	597	0	0	451	1584	4	15	1	0	0	0
1289	599	587	600	604	503	829	4	0	0	0	0	0
1290	600	594	601	0	3	0	4	0	0	0	0	0
1291	601	600	602	0	453	544	4	0	0	0	0	0
1292	602	601	603	0	453	616	4	25	1	0	0	0
1293	603	602	0	0	453	902	4	15	1	0	0	0
1294	604	599	605	0	503	533	4	15	1	0	0	0
1295	605	604	608	0	503	427	4	0	0	0	47	0
1296	608	605	609	611	503	524	4	0	0	0	0	0
1297	609	608	610	0	3	0	4	0	0	0	0	0
1298	610	609	0	0	453	655	4	25	1	0	0	0
1299	611	608	612	0	501	622	4	0	0	0	0	0
1300	612	611	613	0	503	610	4	15	1	0	0	0
1301	613	612	614	624	503	610	4	0	0	0	0	0
1302	614	613	615	0	2	0	4	0	0	0	0	0
1303	615	614	616	0	453	610	4	0	0	0	0	0
1304	616	615	617	0	453	594	4	25	1	0	0	0
1305	617	616	618	0	453	646	4	15	1	0	0	0
1306	618	617	619	0	453	1143	4	25	1	0	0	0
1307	619	618	620	0	453	1137	4	10	1	0	0	0
1308	620	619	621	622	453	491	4	0	0	0	0	0

1309	621	620	0	0	451	747	4	25	1	0	0	0
1310	622	620	0	0	453	1335	4	25	1	0	0	0
1311	624	613	625	0	503	512	4	15	1	0	0	0
1312	625	624	0	0	503	689	4	10	1	0	0	0
1313	630	344	631	0	101	1539	4	0	0	0	0	0
1314	631	630	632	670	127	503	4	0	0	0	0	0
1315	632	631	633	0	127	1277	4	15	2	0	0	0
1316	633	632	634	0	113	1155	4	10	3	0	0	0
1317	634	633	635	0	114	686	4	0	0	0	0	0
1318	635	634	636	654	113	1262	4	0	0	0	0	0
1319	636	635	637	0	129	838	4	25	3	0	0	0
1320	637	636	638	0	105	808	4	0	0	0	0	0
1321	638	637	649	0	106	878	4	40	1	0	0	0
1322	639	649	640	0	2	0	4	0	0	0	0	0
1323	640	639	641	0	455	558	4	0	0	0	0	0
1324	641	640	642	0	456	582	4	25	1	0	0	0
1325	642	641	643	0	457	427	4	0	0	0	0	0
1326	643	642	644	0	455	756	4	0	0	0	0	0
1327	644	643	645	0	456	933	4	25	1	0	0	0
1328	645	644	646	0	451	509	4	0	0	0	0	0
1329	646	645	647	0	452	942	4	25	1	0	0	0
1330	647	646	648	0	451	664	4	10	1	0	0	0
1331	648	647	0	0	451	610	4	10	1	0	0	0
1332	649	638	650	639	129	732	4	40	1	0	0	0
1333	650	649	651	653	126	732	4	40	1	0	0	0
1334	651	650	652	0	1	0	4	0	0	0	0	0
1335	652	651	0	0	305	625	4	25	1	0	0	0
1336	653	650	0	0	129	1488	4	15	3	0	0	0
1337	654	635	655	0	126	914	4	15	3	0	0	0
1338	655	654	656	0	129	2593	4	15	3	0	0	0
1339	656	655	657	0	3	0	4	0	0	0	0	0
1340	657	656	658	0	305	1097	4	10	1	0	0	0
1341	658	657	659	0	305	2160	4	10	0	0	0	0
1342	659	658	660	0	325	2880	4	0	0	0	0	0
1343	660	659	661	0	325	1992	4	10	1	0	0	0
1344	661	660	662	0	325	1776	4	10	1	0	0	0
1345	662	661	0	0	326	1200	4	10	1	0	0	0
1346	670	631	671	0	113	3783	4	0	0	0	0	0
1347	671	670	672	830	101	1198	4	0	0	0	56	0
1348	672	671	673	0	101	847	4	15	1	0	0	0
1349	673	672	674	678	101	1213	4	0	0	0	0	0
1350	674	673	675	0	3	0	4	0	0	0	0	0
1351	675	674	0	0	451	1392	4	25	1	0	0	0
1352	678	673	679	688	129	622	4	0	0	0	0	0
1353	679	678	680	0	1	0	4	0	0	0	0	0
1354	680	679	681	0	305	396	4	15	1	0	0	0
1355	681	680	682	0	309	732	4	0	0	0	0	0
1356	682	681	683	0	305	732	4	25	1	0	0	0
1357	683	682	684	0	305	1676	4	25	1	0	0	0
1358	684	683	685	687	305	808	4	0	0	0	0	0
1359	685	684	686	0	309	549	4	25	1	0	0	0
1360	686	685	0	0	305	579	4	25	1	0	0	0

1361	687	684	0	0	301	366	4	15	1	0	0	0
1362	688	678	691	0	129	753	4	0	0	0	47	0
1363	691	688	692	701	129	643	4	0	0	0	0	0
1364	701	691	702	695	101	0	4	0	0	0	0	0
1365	692	691	0	0	101	488	4	0	0	0	48	0
1366	695	701	696	0	1	0	4	0	0	0	0	0
1367	696	695	697	0	305	622	4	10	1	0	0	0
1368	697	696	698	0	305	625	4	0	0	0	0	0
1369	698	697	699	0	307	390	4	15	1	0	0	0
1370	699	698	700	0	307	448	4	15	1	0	0	0
1371	700	699	0	0	307	1143	4	25	1	0	0	0
1372	702	701	703	0	129	701	4	0	0	0	0	0
1373	703	702	704	0	131	701	4	40	3	0	0	0
1374	704	703	705	0	131	707	4	0	0	0	0	0
1375	705	704	706	780	129	899	4	0	0	0	0	0
1376	706	705	707	786	177	528	4	15	2	0	0	0
1377	707	706	708	0	177	1200	4	25	2	0	0	0
1378	708	707	709	0	177	1524	4	25	2	0	0	0
1379	709	708	710	715	177	544	4	0	0	0	0	0
1380	710	709	711	0	1	0	4	0	0	0	0	0
1381	711	710	712	0	301	823	4	0	0	0	0	0
1382	712	711	713	0	303	1200	4	15	1	0	0	0
1383	713	712	0	0	303	524	4	10	1	0	0	0
1384	715	709	716	0	503	1024	4	15	1	0	0	0
1385	716	715	717	0	503	1008	4	25	1	0	0	0
1386	717	716	718	0	503	1214	4	0	0	0	48	0
1387	718	717	719	0	503	381	4	25	1	0	0	0
1388	719	718	720	725	503	1402	4	0	0	0	0	0
1389	720	719	721	0	2	0	4	0	0	0	0	0
1390	721	720	722	0	451	671	4	0	0	0	0	0
1391	722	721	723	0	453	579	4	25	1	0	0	0
1392	723	722	724	0	453	1745	4	40	1	0	0	0
1393	724	723	0	0	453	579	4	0	0	0	0	0
1394	725	719	726	0	627	610	4	15	1	0	0	0
1395	726	725	727	732	627	381	4	0	0	0	0	0
1396	727	726	728	0	1	0	4	0	0	0	0	0
1397	728	727	729	0	453	1204	4	25	1	0	0	0
1398	729	728	730	0	453	1173	4	40	1	0	0	0
1399	730	729	0	0	453	1554	4	40	1	0	0	0
1400	732	726	733	0	627	610	4	15	1	0	0	0
1401	733	732	734	0	627	1890	4	25	1	0	0	0
1402	734	733	735	736	627	720	4	0	0	0	0	0
1403	735	734	0	0	627	823	4	10	1	0	0	0
1404	736	734	739	0	527	610	4	0	0	0	47	0
1405	739	736	740	0	527	600	4	25	1	0	0	0
1406	740	739	741	743	627	672	4	0	0	0	0	0
1407	741	740	742	0	1	0	4	0	0	0	0	0
1408	742	741	0	0	453	732	4	25	1	0	0	0
1409	743	740	744	0	627	640	4	15	1	0	0	0
1410	744	743	745	748	525	1402	4	0	0	0	0	0
1411	745	744	746	0	1	0	4	0	0	0	0	0
1412	746	745	747	0	453	579	4	25	1	0	0	0

1413	747	746	0	0	453	716	4	0	0	0	0	0
1414	748	744	749	0	627	664	4	0	0	0	0	0
1415	749	748	750	770	625	701	4	0	0	0	0	0
1416	750	749	751	0	527	610	4	40	1	0	0	0
1417	751	750	752	753	627	1585	4	0	0	0	0	0
1418	752	751	0	0	627	792	4	25	1	0	0	0
1419	753	751	754	0	625	732	4	25	2	0	0	0
1420	754	753	755	758	627	500	4	0	0	0	0	0
1421	755	754	756	0	2	0	4	0	0	0	0	0
1422	756	755	0	0	451	655	4	25	1	0	0	0
1423	758	754	759	0	627	664	4	25	2	0	0	0
1424	759	758	760	765	627	722	4	0	0	0	0	0
1425	760	759	761	0	1	0	4	0	0	0	0	0
1426	761	760	762	0	451	625	4	0	0	0	0	0
1427	762	761	763	0	453	732	4	25	1	0	0	0
1428	763	762	764	0	453	1854	4	25	1	0	0	0
1429	764	763	0	0	453	1311	4	10	1	0	0	0
1430	765	759	766	0	527	1008	4	25	3	0	0	0
1431	766	765	769	0	527	661	4	0	0	0	48	0
1432	769	766	0	0	527	720	4	25	1	0	0	0
1433	770	749	771	0	2	0	4	0	0	0	0	0
1434	771	770	772	0	451	1350	4	0	0	0	0	0
1435	772	771	773	774	451	625	4	0	0	0	0	0
1436	773	772	0	0	453	528	4	15	1	0	0	0
1437	774	772	0	0	453	1008	4	25	1	0	0	0
1438	780	705	781	0	2	0	4	0	0	0	0	0
1439	781	780	782	0	451	823	4	25	1	0	0	0
1440	782	781	783	0	457	1875	4	25	1	0	0	0
1441	783	782	784	0	457	789	4	25	1	0	0	0
1442	784	783	785	0	457	1353	4	25	1	0	0	0
1443	785	784	0	0	457	762	4	0	0	0	0	0
1444	786	706	787	792	127	546	4	0	0	0	0	0
1445	787	786	788	0	2	0	4	0	0	0	0	0
1446	788	787	789	0	455	655	4	0	0	0	0	0
1447	789	788	790	0	457	357	4	25	1	0	0	0
1448	790	789	791	0	457	1414	4	25	1	0	0	0
1449	791	790	0	0	453	707	4	0	0	0	0	0
1450	792	786	793	0	127	530	4	25	3	0	0	0
1451	793	792	794	0	129	1295	4	25	3	0	0	0
1452	794	793	795	0	129	1838	4	25	1	0	0	0
1453	795	794	796	798	127	411	4	0	0	0	0	0
1454	796	795	0	0	127	720	4	15	1	0	0	0
1455	798	795	799	817	127	646	4	0	0	0	0	0
1456	799	798	800	0	2	0	4	0	0	0	0	0
1457	800	799	801	0	451	640	4	15	1	0	0	0
1458	801	800	802	805	453	640	4	0	0	0	0	0
1459	802	801	803	0	451	823	4	0	0	0	0	0
1460	803	802	804	0	453	671	4	25	1	0	0	0
1461	804	803	0	0	453	1104	4	25	1	0	0	0
1462	805	801	806	0	451	762	4	0	0	0	0	0
1463	806	805	807	0	453	808	4	25	1	0	0	0
1464	807	806	808	0	453	768	4	0	0	0	0	0

1465	808	807	809	0	451	786	4	0	0	0	0	0
1466	809	808	810	0	461	610	4	15	1	0	0	0
1467	810	809	811	0	455	610	4	0	0	0	0	0
1468	811	810	812	0	453	610	4	25	1	0	0	0
1469	812	811	813	0	453	823	4	0	0	0	0	0
1470	813	812	814	815	451	671	4	0	0	0	0	0
1471	814	813	0	0	453	701	4	25	1	0	0	0
1472	815	813	816	0	451	326	4	0	0	0	0	0
1473	816	815	0	0	453	634	4	25	1	0	0	0
1474	817	798	818	0	127	747	4	25	1	0	0	0
1475	818	817	819	0	129	1158	4	0	0	0	0	0
1476	819	818	820	0	101	652	4	0	0	0	0	0
1477	820	819	821	0	127	820	4	15	2	0	0	0
1478	821	820	822	0	127	799	4	0	0	0	0	0
1479	822	821	823	0	101	762	4	25	1	0	0	0
1480	823	822	824	0	127	488	4	0	0	0	0	0
1481	824	823	825	0	101	1241	4	15	2	0	0	0
1482	825	824	826	0	127	1951	4	0	0	0	0	0
1483	826	825	827	0	101	564	4	0	0	0	51	0
1484	827	826	0	0	101	1214	4	0	0	0	46	0
1485	830	671	831	833	101	1488	4	0	0	0	0	0
1486	831	830	832	0	1	0	4	0	0	0	0	0
1487	832	831	0	0	453	777	4	25	1	0	0	0
1488	833	830	1029	1012	101	576	4	15	3	0	55	0
1489	1029	833	0	0	101	0	4	0	0	0	0	0
1490	1012	833	0	1013	101	1296	4	0	0	0	0	0
1491	1013	1012	835	834	101	624	4	0	0	0	0	0
1492	834	1013	1750	1014	101	0	4	0	0	0	0	0
1493	1014	834	1754	838	101	0	4	0	0	0	0	0
1494	835	1013	836	0	1	0	4	0	0	0	0	0
1495	836	835	837	0	451	594	4	15	1	0	0	0
1496	837	836	0	0	453	360	4	25	1	0	0	0
1497	838	1014	839	0	101	528	4	0	0	0	0	0
1498	839	838	840	1515	101	1248	4	0	0	0	0	0
1499	1515	839	1508	0	101	768	4	0	0	0	0	0
1500	1508	1515	0	0	903	960	4	0	0	0	43	0
1501	840	839	841	0	101	2807	4	10	2	0	0	0
1502	841	840	842	844	101	3847	4	0	0	0	0	0
1503	842	841	843	0	101	691	4	0	0	0	0	0
1504	843	842	1160	0	101	625	4	0	0	0	0	0
1505	844	841	845	0	525	701	4	15	3	0	0	0
1506	845	844	846	0	527	686	4	40	3	0	0	0
1507	846	845	847	849	527	1515	4	0	0	0	0	0
1508	847	846	848	0	2	0	4	0	0	0	0	0
1509	848	847	0	0	451	747	4	25	1	0	0	0
1510	849	846	850	0	527	518	4	25	3	0	0	0
1511	850	849	851	854	527	716	4	0	0	0	0	0
1512	851	850	852	0	3	0	4	0	0	0	0	0
1513	852	851	853	0	453	1521	4	25	1	0	0	0
1514	853	852	0	0	453	585	4	0	0	0	0	0
1515	854	850	855	858	527	0	4	0	0	0	0	0
1516	855	854	856	0	3	0	4	0	0	0	0	0

1517	856	855	857	0	451	610	4	0	0	0	0	0
1518	857	856	0	0	453	644	4	25	1	0	0	0
1519	858	854	859	0	527	814	4	25	3	0	0	0
1520	859	858	860	865	527	604	4	0	0	0	0	0
1521	860	859	861	0	2	0	4	0	0	0	0	0
1522	861	860	862	0	453	1402	4	25	1	0	0	0
1523	862	861	863	0	453	2185	4	25	1	0	0	0
1524	863	862	864	0	453	1768	4	25	1	0	0	0
1525	864	863	0	0	453	658	4	0	0	0	0	0
1526	865	859	866	870	527	442	4	0	0	0	0	0
1527	866	865	867	0	1	0	4	0	0	0	0	0
1528	867	866	868	0	453	1524	4	25	1	0	0	0
1529	868	867	869	0	453	480	4	25	1	0	0	0
1530	869	868	0	0	453	2271	4	25	1	0	0	0
1531	870	865	871	0	527	701	4	25	1	0	0	0
1532	871	870	872	0	527	640	4	15	3	0	0	0
1533	872	871	0	0	527	1149	4	0	0	0	47	0
1534	875	1160	876	0	101	558	4	0	0	0	0	0
1535	876	875	877	879	125	1152	4	0	0	0	0	0
1536	877	876	878	0	2	0	4	0	0	0	0	0
1537	878	877	0	0	453	747	4	10	1	0	0	0
1538	879	876	881	880	525	644	4	0	0	0	0	0
1539	880	879	0	0	527	728	4	25	1	0	0	0
1540	881	879	882	885	525	905	4	0	0	0	0	0
1541	882	881	883	0	2	0	4	0	0	0	0	0
1542	883	882	884	0	453	564	4	25	1	0	0	0
1543	884	883	0	0	453	457	4	0	0	0	0	0
1544	885	881	886	0	525	628	4	15	3	0	0	0
1545	886	885	887	0	527	981	4	0	0	0	0	0
1546	887	886	889	888	525	655	4	0	0	0	0	0
1547	888	887	0	0	527	747	4	25	3	0	0	0
1548	889	887	890	0	527	1344	4	0	0	0	0	0
1549	1509	891	1510	0	2	0	4	0	0	0	0	0
1550	1510	1509	1511	0	453	1134	4	25	1	0	0	0
1551	1511	1510	1512	1514	453	912	4	0	0	0	0	0
1552	1512	1511	1513	0	453	672	4	40	1	0	0	0
1553	1513	1512	0	0	453	672	4	0	0	0	0	0
1554	1514	1511	0	0	453	2017	4	25	1	0	0	0
1555	890	889	891	0	627	1200	4	75	1	0	0	0
1556	891	890	892	1509	527	1067	4	15	1	0	0	0
1557	892	891	904	893	527	0	4	0	0	0	0	0
1558	893	892	894	0	2	0	4	0	0	0	0	0
1559	894	893	895	0	453	579	4	25	1	0	0	0
1560	895	894	896	0	451	3121	4	15	1	0	0	0
1561	896	895	897	898	451	0	4	0	0	0	0	0
1562	897	896	0	0	453	768	4	25	1	0	0	0
1563	898	896	0	0	453	1872	4	25	1	0	0	0
1564	904	892	905	0	527	351	4	0	0	0	0	0
1565	905	904	906	0	525	792	4	0	0	0	0	0
1566	906	905	907	0	527	747	4	25	1	0	0	0
1567	907	905	908	911	527	707	4	0	0	0	0	0
1568	908	90	909	0	1	0	4	0	0	0	0	0

1569	909	908	910	0	451	732	4	10	1	0	0	0
1570	910	909	0	0	453	1969	4	25	1	0	0	0
1571	911	907	912	0	627	747	4	15	1	0	0	0
1572	912	911	913	0	627	777	4	15	3	0	0	0
1573	913	912	914	0	627	472	4	0	0	0	0	0
1574	914	913	915	0	227	914	4	25	1	0	0	0
1575	915	914	916	923	627	1463	4	0	0	0	0	0
1576	916	915	917	0	2	0	4	0	0	0	0	0
1577	917	916	918	0	453	329	4	25	1	0	0	0
1578	918	917	919	0	453	1686	4	25	1	0	0	0
1579	919	918	920	0	453	1839	4	25	1	0	0	0
1580	920	919	921	0	453	1296	4	0	0	0	0	0
1581	921	920	0	0	453	1244	4	0	0	0	0	0
1582	923	915	924	0	627	624	4	25	1	0	0	0
1583	924	923	925	927	627	716	4	0	0	0	0	0
1584	925	924	926	0	1	0	4	0	0	0	0	0
1585	926	925	0	0	453	912	4	40	1	0	0	0
1586	927	924	930	0	227	701	4	0	0	0	47	0
1587	930	927	931	0	527	686	4	25	3	0	0	0
1588	931	930	932	934	627	640	4	0	0	0	0	0
1589	932	931	933	0	3	0	4	0	0	0	0	0
1590	933	932	0	0	451	427	4	10	1	0	0	0
1591	934	931	935	0	627	671	4	25	1	0	0	0
1592	935	934	936	0	627	1533	4	25	3	0	0	0
1593	936	935	0	0	627	1008	4	0	0	0	0	0
1594	1019	1018	940	0	975	750	4	0	0	0	0	0
1595	1018	0	1019	0	975	0	4	0	0	0	60	0
1596	940	1019	941	0	102	0	4	15	0	0	0	0
1597	941	940	942	0	102	2208	4	10	1	0	0	0
1598	942	941	943	0	102	6528	4	0	0	0	0	0
1599	943	942	944	0	102	1440	4	10	2	0	0	0
1600	944	943	945	0	102	1152	4	15	1	0	0	0
1601	945	944	946	0	102	1056	4	15	3	0	0	0
1602	946	945	1	0	101	672	4	0	0	0	0	0
1603	1160	843	875	1161	101	1862	4	0	0	0	0	0
1604	1161	1160	1162	1165	101	1920	4	0	0	0	0	0
1605	1162	1161	0	0	101	640	4	0	0	0	0	0
1606	1165	1161	1166	0	101	1393	4	10	3	0	46	0
1607	1166	1165	1167	0	101	624	4	10	2	0	0	0
1608	1167	1166	1169	1110	101	732	4	15	2	0	0	0
1609	1110	1167	0	0	903	1524	4	0	0	0	42	0
1610	1168	1167	1169	1172	101	1311	4	0	0	0	0	0
1611	1169	1168	0	0	101	700	4	0	0	0	44	0
1612	1172	1168	1173	1176	101	1200	4	0	0	0	0	0
1613	1173	1172	0	0	101	1392	4	0	0	0	46	0
1614	1176	1172	1177	0	101	1750	4	0	0	0	43	0
1615	1303	1177	1304	0	3	0	4	0	0	0	0	0
1616	1304	1303	0	0	451	672	4	10	1	0	0	0
1617	1177	1176	1305	1303	101	1750	4	0	0	0	0	0
1618	1305	1177	1178	1215	101	0	4	0	0	0	0	0
1619	1178	1305	1179	1182	105	3793	4	0	0	0	0	0
1620	1179	1178	0	0	101	1008	4	0	0	0	46	0

1621	1182	1178	1183	0	109	1200	4	25	3	0	0	0
1622	1183	1182	1184	0	109	1400	4	15	2	0	0	0
1623	1184	1183	1187	0	109	2400	4	0	0	0	47	0
1624	1187	1184	1188	1030	109	610	4	0	0	0	0	0
1625	1030	1187	0	0	109	0	4	0	0	0	0	0
1626	1188	1187	1189	0	3	0	4	0	0	0	0	0
1627	1189	1188	1190	0	315	610	4	0	0	0	0	0
1628	1190	1189	1191	0	315	2250	4	0	0	0	0	0
1629	1191	1190	1192	1193	313	2250	4	0	0	0	0	0
1630	1193	1191	0	0	315	2209	4	15	1	0	0	0
1631	1192	1191	1194	0	318	610	4	10	1	0	0	0
1632	1194	1192	1195	0	313	3601	4	25	1	0	0	0
1633	1195	1194	1196	0	318	2113	4	15	1	0	0	0
1634	1196	1195	1197	1201	317	3169	4	0	0	0	0	0
1635	1197	1196	1198	0	317	5973	4	10	1	0	0	0
1636	1198	1197	1199	1200	317	1488	4	5	1	0	0	0
1637	1199	1198	0	0	317	1632	4	15	1	0	0	0
1638	1200	1198	0	0	317	912	4	5	1	0	0	0
1639	1201	1196	1202	0	317	1728	4	10	1	0	0	0
1640	1202	1201	1203	0	317	3985	4	10	1	0	0	0
1641	1203	1202	1204	1205	317	4708	4	0	0	0	0	0
1642	1204	1203	0	0	317	610	4	25	1	0	0	0
1643	1205	1203	1206	1208	317	610	4	0	0	0	0	0
1644	1206	1205	1207	1209	317	2929	4	10	1	0	0	0
1645	1207	1206	0	0	317	1151	4	0	0	0	0	0
1646	1208	1205	0	0	317	4081	4	0	0	0	0	0
1647	1209	1206	1210	0	317	610	4	10	1	0	0	0
1648	1210	1209	0	0	317	2977	4	15	1	0	0	0
1649	1215	1305	1216	1222	101	3937	4	0	0	0	0	0
1650	1216	1215	1217	0	501	893	4	0	0	0	44	0
1651	1217	1216	1218	1220	501	700	4	0	0	0	0	0
1652	1220	1217	0	0	903	310	4	0	0	0	42	0
1653	1218	1217	1219	1221	501	700	4	0	0	0	0	0
1654	1221	1218	0	0	903	310	4	0	0	0	42	0
1655	1219	1218	0	0	201	576	4	0	0	0	0	0
1656	1222	1215	0	0	101	1344	4	0	0	0	56	0
1657	1300	101	1301	0	525	0	4	0	0	0	0	0
1658	1301	1300	1302	0	525	0	4	0	0	0	0	0
1659	1302	1301	0	0	903	1100	4	0	0	0	41	0
1660	1655	351	1011	1787	101	0	4	0	0	0	0	0
1661	1011	1655	1656	0	1	0	4	0	0	0	0	0
1662	1656	1011	1657	0	905	1774	4	25	1	0	0	0
1663	1657	1656	1658	0	905	601	4	25	1	0	0	0
1664	1658	1657	1659	0	905	1058	4	25	1	0	0	0
1665	1659	1658	1660	0	905	402	4	25	1	0	0	0
1666	1660	1659	1661	0	905	768	4	25	1	0	0	0
1667	1661	1660	1662	0	905	655	4	50	1	0	0	0
1668	1662	1661	1663	0	905	905	4	0	0	0	0	0
1669	1663	1662	1664	0	905	1308	4	50	1	0	0	0
1670	1664	1663	1665	0	905	677	4	25	1	0	0	0
1671	1665	1664	1666	0	905	677	4	25	1	0	0	0
1672	1666	1665	1667	0	905	1332	4	25	1	0	0	0

1673	1667	1666	1668	0	905	1317	4	25	1	0	0	0
1674	1668	1667	1669	0	905	1317	4	25	1	0	0	0
1675	1669	1668	1670	0	905	677	4	25	1	0	0	0
1676	1670	1669	1671	0	905	753	4	25	1	0	0	0
1677	1671	1670	1672	0	905	598	4	25	1	0	0	0
1678	1672	1671	1673	0	905	655	4	25	1	0	0	0
1679	1673	1672	1674	0	905	677	4	25	1	0	0	0
1680	1674	1673	1785	0	905	677	4	25	1	0	0	0
1681	1785	1674	1786	0	905	1287	4	50	1	0	0	0
1682	1786	1785	1789	0	905	1439	4	25	1	0	0	0
1683	1789	1786	1790	0	905	738	4	25	1	0	0	0
1684	1790	1789	1791	0	905	738	4	25	1	0	0	0
1685	1791	1790	1792	0	905	1012	4	25	1	0	0	0
1686	1792	1791	1793	0	905	662	4	25	1	0	0	0
1687	1793	1792	1794	0	905	982	4	25	1	0	0	0
1688	1794	1793	0	0	905	662	4	0	0	0	0	0
1689	1015	357	1690	1675	101	0	4	0	0	0	0	0
1690	1675	1015	1676	0	3	0	4	0	0	0	0	0
1691	1676	1675	1677	0	905	2756	4	25	1	0	0	0
1692	1677	1676	1678	0	905	820	4	25	1	0	0	0
1693	1678	1677	1679	0	905	951	4	25	1	0	0	0
1694	1679	1678	1680	0	905	1866	4	25	1	0	0	0
1695	1680	1679	1681	0	905	1134	4	25	1	0	0	0
1696	1681	1680	1682	0	905	1058	4	25	1	0	0	0
1697	1682	1681	1683	0	905	1287	4	25	1	0	0	0
1698	1683	1682	0	0	905	1287	4	25	1	0	0	0
1699	1690	1015	1691	0	1	0	4	0	0	0	0	0
1700	1691	1690	1692	0	905	561	4	25	1	0	0	0
1701	1692	1691	1693	0	905	710	4	25	1	0	0	0
1702	1693	1692	1694	0	905	631	4	25	1	0	0	0
1703	1694	1693	1695	0	905	1034	4	25	1	0	0	0
1704	1695	1694	1696	0	905	637	4	50	1	0	0	0
1705	1696	1695	1697	0	905	817	4	25	1	0	0	0
1706	1697	1696	1698	0	905	735	4	25	1	0	0	0
1707	1698	1697	1699	0	905	829	4	25	1	0	0	0
1708	1699	1698	1700	0	905	1491	4	50	1	0	0	0
1709	1700	1699	1701	0	905	655	4	25	1	0	0	0
1710	1701	1700	1702	0	905	756	4	50	1	0	0	0
1711	1702	1701	0	0	905	677	4	50	0	0	0	0
1712	1703	1787	1704	0	905	2790	4	50	1	0	0	0
1713	1787	1655	1703	0	3	0	4	0	0	0	0	0
1714	1704	1703	1705	0	905	732	4	25	1	0	0	0
1715	1705	1704	1706	0	905	829	4	25	1	0	0	0
1716	1706	1705	1707	0	905	851	4	50	1	0	0	0
1717	1707	1706	1708	0	905	915	4	25	1	0	0	0
1718	1708	1707	1709	0	905	655	4	15	1	0	0	0
1719	1709	1708	1788	0	905	698	4	25	1	0	0	0
1720	1788	1709	1684	0	905	753	4	0	0	0	0	0
1721	1684	1788	1530	0	905	1287	4	0	0	0	0	0
1722	1530	1684	1531	0	905	1663	4	0	0	0	0	0
1723	1531	1530	0	0	905	951	4	0	0	0	0	0
1724	1710	433	1711	0	127	0	4	0	0	0	0	0

1725	1711	1710	1712	0	3	0	4	0	0	0	0	0
1726	1712	1711	1713	0	905	1424	4	25	1	0	0	0
1727	1713	1712	1714	0	905	1085	4	25	1	0	0	0
1728	1714	1713	1715	0	905	1637	4	25	1	0	0	0
1729	1715	1714	1716	0	905	1622	4	25	1	0	0	0
1730	1716	1715	0	0	905	1338	4	25	1	0	0	0
1731	1770	434	1717	0	3	0	4	0	0	0	0	0
1732	1717	1770	1718	0	905	860	4	25	1	0	0	0
1733	1718	1717	1719	0	905	1256	4	50	1	0	0	0
1734	1719	1718	1771	0	905	1256	4	75	1	0	0	0
1735	1771	1719	0	0	905	1274	4	50	1	0	0	0
1736	1720	506	1721	1775	931	1400	4	25	1	0	0	0
1737	1775	1720	1776	0	1	0	4	0	0	0	0	0
1738	1776	1775	0	0	905	631	4	25	1	0	0	0
1739	1721	1720	1729	0	931	3452	4	0	0	0	0	0
1740	1722	1729	1723	0	2	0	4	0	0	0	0	0
1741	1723	1722	1724	0	905	814	4	25	1	0	0	0
1742	1724	1723	1725	0	905	655	4	25	1	0	0	0
1743	1725	1724	1726	0	905	1027	4	25	1	0	0	0
1744	1726	1725	1727	0	905	387	4	25	1	0	0	0
1745	1727	1726	1728	0	905	747	4	25	1	0	0	0
1746	1728	1727	0	0	905	826	4	25	1	0	0	0
1747	1729	1721	1730	1722	903	842	4	25	2	0	0	0
1748	1730	1729	1731	1000	903	1723	4	25	1	0	0	0
1749	1000	1730	1010	1737	903	0	4	0	0	0	0	0
1750	1010	1000	0	1001	2	0	4	0	0	0	0	0
1751	1001	1010	1002	0	905	841	4	25	1	0	0	0
1752	1002	1001	1003	0	905	1368	4	25	1	0	0	0
1753	1003	1002	1004	0	905	1259	4	25	1	0	0	0
1754	1004	1003	1005	0	905	1119	4	25	1	0	0	0
1755	1005	1004	1006	0	905	2622	4	25	1	0	0	0
1756	1006	1005	1007	0	905	1049	4	25	1	0	0	0
1757	1007	1006	1008	0	905	677	4	25	1	0	0	0
1758	1008	1007	0	0	905	677	4	25	1	0	0	0
1759	1731	1730	1732	0	1	0	4	0	0	0	0	0
1760	1732	1731	1733	0	905	571	4	25	1	0	0	0
1761	1733	1732	1734	0	905	774	4	25	1	0	0	0
1762	1734	1733	1735	1780	905	1924	4	0	0	0	0	0
1763	1780	1734	1781	0	905	335	4	0	0	0	0	0
1764	1781	1780	0	0	905	260	4	0	0	0	0	0
1765	1735	1734	1736	0	905	791	4	25	1	0	0	0
1766	1736	1735	1520	0	905	381	4	25	1	0	0	0
1767	1520	1736	1521	0	905	721	4	25	1	0	0	0
1768	1521	1520	1522	0	905	1171	4	25	1	0	0	0
1769	1522	1521	0	0	905	402	4	25	1	0	0	0
1770	1737	1000	1738	0	3	0	4	0	0	0	0	0
1771	1738	1737	1739	0	905	937	4	25	1	0	0	0
1772	1739	1738	1740	0	905	1244	4	25	1	0	0	0
1773	1740	1739	1741	0	905	774	4	25	1	0	0	0
1774	1741	1740	1742	0	905	753	4	25	1	0	0	0
1775	1742	1741	0	0	905	430	4	0	0	0	0	0
1776	1750	834	1751	0	101	0	4	0	0	0	0	0

1777	1751	1750	1752	0	2	0	4	0	0	0	0	0
1778	1752	1751	1753	0	905	1143	4	75	1	0	0	0
1779	1753	1752	0	0	905	2354	4	50	1	0	0	0
1780	1754	1014	1755	0	3	0	4	0	0	0	0	0
1781	1755	1754	1756	0	905	2445	4	75	1	0	0	0
1782	1756	1755	1757	0	905	3201	4	25	1	0	0	0
1783	1757	1756	0	0	905	1297	4	25	1	0	0	0
1784	1760	422	1761	0	905	979	4	25	1	0	0	0
1785	1761	1760	1762	0	905	1262	4	25	1	0	0	0
1786	1762	1761	1763	0	905	1439	4	50	1	0	0	0
1787	1763	1762	1764	0	905	1256	4	50	1	0	0	0
1788	1764	1763	0	0	905	835	4	75	1	0	0	0
1789	521	520	522	523	401	277	4	0	0	0	0	0
1790	522	521	0	0	401	317	4	25	4	0	0	0
1791	523	521	524	525	401	1475	4	0	0	0	0	0
1792	524	523	0	0	401	451	4	50	4	0	0	0
1793	525	523	526	527	401	1140	4	10	4	0	0	0
1794	526	525	0	0	401	1844	4	25	4	0	0	0
1795	527	525	528	0	401	1018	4	25	4	0	0	0
1796	528	527	529	0	401	902	4	5	4	0	0	0
1797	529	528	530	0	401	616	4	15	4	0	0	0
1798	530	529	531	0	401	671	4	5	4	0	0	0
1799	531	530	532	0	401	2015	4	0	0	0	0	0
1800	532	531	533	0	401	684	4	0	0	0	0	0
1801	533	532	534	535	401	1049	4	5	4	0	0	0
1802	534	533	0	0	401	1104	4	25	4	0	0	0
1803	535	533	536	537	401	3283	4	0	0	0	0	0
1804	536	535	0	0	401	549	4	0	0	0	0	0
1805	537	535	538	0	401	1284	4	15	4	0	0	0
1806	538	537	539	0	401	1622	4	0	0	0	0	0
1807	539	538	540	541	401	509	4	0	0	0	0	0
1808	540	539	0	0	401	2839	4	0	0	0	0	0
1809	541	539	542	0	401	5145	4	0	0	0	0	0
1810	542	541	543	0	401	692	4	10	4	0	0	0
1811	543	542	544	0	401	744	4	0	0	0	0	0
1812	544	543	545	0	401	4450	4	0	0	0	0	0
1813	545	544	546	0	401	1173	4	0	0	0	0	0
1814	546	545	547	0	401	375	4	10	4	0	0	0
1815	547	546	548	0	401	631	4	15	4	0	0	0
1816	548	547	549	0	401	1292	4	0	0	0	0	0
1817	549	548	550	0	401	713	4	10	4	0	0	0
1818	550	549	551	0	401	3082	4	15	4	0	0	0
1819	551	550	552	553	401	1478	4	15	4	0	0	0
1820	552	551	0	0	401	512	4	15	4	0	0	0
1821	553	551	554	0	401	856	4	0	0	0	0	0
1822	554	553	555	0	401	1448	4	0	0	0	0	0
1823	555	554	556	0	401	750	4	10	4	4	0	0
1824	556	555	557	0	401	1506	4	0	0	0	0	0
1825	557	556	558	0	401	1195	4	10	4	0	0	0
1826	558	557	559	0	401	972	4	15	4	0	0	0
1827	559	558	560	0	401	4712	4	15	4	0	0	0
1828	560	559	561	0	401	610	4	0	0	0	0	0

1829	561	560	562	0	401	707	4	40	4	0	0	0
1830	562	561	0	0	401	710	4	0	0	0	0	0
1831	373	372	374	0	401	619	4	0	0	0	0	0
1832	374	373	375	0	401	1457	4	15	4	0	0	0
1833	375	374	376	0	401	2840	4	10	4	0	0	0
1834	376	375	377	0	401	344	4	10	4	0	0	0
1835	378	377	379	0	401	1387	4	10	4	0	0	0
1836	377	376	378	0	401	442	4	0	0	0	0	0
1837	379	378	380	0	401	878	4	10	4	0	0	0
1838	380	379	381	0	401	1922	4	15	4	0	0	0
1839	381	380	382	0	401	529	4	0	0	0	0	0
1840	382	381	383	0	401	869	4	5	4	0	0	0
1841	383	382	384	385	401	991	4	0	0	0	0	0
1842	384	383	0	0	401	677	4	10	4	0	0	0
1843	385	383	386	0	401	610	4	10	4	0	0	0
1844	386	385	387	0	401	1228	4	10	4	0	0	0
1845	387	386	388	0	401	3149	4	10	4	0	0	0
1846	388	387	389	0	401	722	4	40	4	0	0	0
1847	389	388	390	0	401	847	4	10	4	0	0	0
1848	390	389	391	0	401	510	4	0	0	0	0	0
1849	391	390	392	393	401	1430	4	5	4	0	0	0
1850	392	391	0	0	401	732	4	5	4	0	0	0
1851	393	391	394	0	401	4804	4	0	0	0	0	0
1852	394	393	395	397	401	4	4	0	0	0	0	0
1853	395	394	396	413	401	4	4	0	0	0	0	0
1854	396	395	0	0	401	384	4	0	0	0	0	0
1855	397	394	398	0	401	2283	4	10	4	0	0	0
1856	398	397	399	0	401	1594	4	10	4	0	0	0
1857	399	398	403	400	401	503	4	10	4	0	0	0
1858	400	399	402	401	401	1414	4	0	0	0	0	0
1859	401	400	0	0	401	323	4	0	0	0	0	0
1860	402	400	0	0	401	1097	4	15	4	0	0	0
1961	403	399	405	404	401	2932	4	0	0	0	0	0
1862	404	403	0	0	401	604	4	40	4	0	0	0
1863	405	403	407	406	401	4225	4	0	0	0	0	0
1864	406	405	0	0	401	396	4	15	4	0	0	0
1865	407	405	408	0	401	1219	4	15	4	0	50	0
1866	408	407	409	0	401	4715	4	5	4	0	0	0
1867	409	408	410	411	401	1006	4	0	0	0	0	0
1868	410	409	0	0	401	680	4	15	4	0	0	0
1869	411	409	412	0	401	1347	4	15	4	0	0	0
1870	412	411	0	0	401	881	4	10	4	0	0	0
1871	413	395	414	0	401	838	4	15	4	0	0	0
1872	414	413	415	416	401	1119	4	0	0	0	0	0
1873	415	414	0	0	401	405	4	10	4	0	0	0
1874	416	414	417	0	401	646	4	10	4	0	0	0
1875	417	416	0	0	401	2377	4	10	4	0	0	0

```
1010 101,99 101 0X0C2X
1020 3,1,0,0,1
1040 1
1050 3
1070 .2790,,2798,,0217,,666
1090 1.4096,,6653,,316,3.72E7
1120 -3.667,33.667
1121 0.000,35.0e3
1122 3.667,33.667
1123 .583,27.333
1128 1
1010 103,99 121 0X0002
1020 3,1,0,0,1
1040 1
1050 3
1070 .2790,,2798,,0217,,666
1090 .3547,,3848,,629,3.72E7
1120 -3.667,33.667
1121 0.000,35.063
1122 3.667,33.667
1123 .583,27.333
1128 1
1010 105,99 103 0X0C0X M1
1020 3,2,0,0,1
1040 1
1050 3
1070 .2790,,2798,,0217,,666
1090 .8870,,6547,,398,3.72E7
1091 1.600,,4773,,500,3.72E7
1120 -3.667,33.667
1121 0.000,35.063
1122 3.667,33.667
1123 .583,27.333
1124 .333,22.500
1128 1
1010 107,99 103 0X002X M1
1020 3,2,0,0,1
1040 1
1050 3
1070 .2790,,2798,,0217,,666
1090 1.4096,,6653,,316,3.72E7
1091 1.600,,4773,,500,3.72E7
1120 -3.667,33.667
1121 0.000,35.063
1122 3.667,33.667
1123 .583,27.333
1124 .333,22.500
1128 1
1010 109,99 105 0X0C2X M3
```

```
1020 3,2,0,0,1
1040 1
1050 3
1070 .2790,,2798,,0217,,666
1090 1.4000,,6653,,316,3.72E7
1091 .1000,,3092,2.000,3.72E7
1120 -3.667,33.667
1121 0.000,35.063
1122 3.667,33.667
1123 .583,27.333
1124 .333,22.500
1128 1
1010 111,99 105 0X000X M3
1020 3,2,0,0,1
1040 1
1050 3
1070 .2790,,2798,,0217,,666
1090 .8870,,6547,,398,3.72E7
1091 .1000,,3092,2.000,3.72E7
1120 -3.667,33.667
1121 0.000,35.063
1122 3.667,33.667
1123 .583,27.333
1124 .333,22.500
1128 1
1010 113,99 137 0X0002 MULTI TELEPHONE LINES
1020 3,2,0,0,1
1040 1
1050 3
1070 .2790,,2798,,0217,,666
1090 .3547,,3846,,629,3.72E7
1091 1.600,,4773,,500,3.72E7
1120 -3.667,33.667
1121 0.000,35.063
1122 3.667,33.667
1123 .583,27.333
1124 .333,22.500
1128 1
1010 115,99 125 0X0002 M3
1020 3,2,0,0,1
1040 1
1050 3
1070 .2790,,2798,,0217,,666
1090 .3547,,3846,,629,3.72E7
1091 .1000,,3092,2.000,3.72E7
1120 -3.667,33.667
1121 0.000,35.063
1122 3.667,33.667
1123 .583,27.333
1124 .333,22.500
1128 1
1010 117,99 241 1X000X
```

```
1020 3,1,0,0,1
1040 1
1050 3
1070 .8862,.8870,.0045,.396
1090 .8870,.6547,.398,3.72E7
1120 -1.75,33.25
1121 0.000,40.00
1122 1.75,33.25
1123 .583,27.500
1128 1
1010 119,99 241 0X000X
1020 3,1,0,0,1
1040 1
1050 3
1070 .2790,.2798,.0217,.666
1090 .8870,.6547,.396,3.72E7
1120 -1.75,33.25
1121 0.000,35.000
1122 1.75,33.25
1123 .583,27.500
1128 1
1010 121,99 261 0X0002
1020 3,1,0,0,1
1040 1
1050 3
1070 .2790,.2798,.0217,.666
1090 .3547,.3845,.629,3.72E7
1120 -1.75,33.25
1121 0.000,35.000
1122 1.75,33.333
1123 .583,27.500
1128 1
1010 125,99 195 0X0010 M3
1020 3,3,0,0,1
1040 1
1050 3
1070 .2790,.2798,.0217,.666
1090 1.335,.6830,.4330,3.72E7
1091 .8870,.6547,.398,3.72E7
1092 .1000,.3092,2.000,3.72E7
1120 .670,32.810
1121 1.000,32.290
1122 1.333,32.810
1123 1.000,33.333
1124 .583,33.292
1125 .333,23.458
1128 1
1010 127,99 421 2X000X
1020 1,2,0,0,1
1040 1
1050 1
1070 1.4076,1.4098,.00510,.316
```

```
1090 1.4098,,6653,,316,3.72E7
1091 .8870,,6547,,398,3.72E7
1120 -3.667,33.667
1121 3.667,33.667
1122 .5833,27.333
1128 1
1010 129,99 601 1X000X
1020 1,1,0,0,1
1040 1
1050 1
1070 .8862,,8870,,0045,,398
1090 .8870,,6547,,398,3.72E7
1120 0.000,35.000
1121 .583,27.333
1128 1
1010 131,99 601 1X002X
1020 1,1,0,0,1
1040 1
1050 1
1070 .8862,,8870,,0045,,398
1090 1.4098,,6653,,316,3.72E7
1120 0.000,35.000
1121 .583,27.333
1128 1
1010 133,99 601 2X002X
1020 1,1,0,0,1
1040 1
1050 1
1070 1.4076,1.4098,,00510,,316
1090 1.4098,,6653,,316,3.72E7
1120 0.000,35.000
1121 .583,27.333
1128 1
1010 135,99 621 1X0002
1020 1,1,0,0,1
1040 1
1050 1
1070 .8862,,8870,,0045,,398
1090 .3547,,3848,,629,3.72E7
1120 0.000,35.000
1121 .583,27.333
1128 1
1010 137,99 625 1X0002 M3
1020 1,2,0,0,1
1040 1
1050 1
1070 .8862,,8870,,0045,,398
1090 .3547,,3848,,629,3.72E7
1091 .1000,,3092,2.000,3.72E7
1120 0.000,35.000
1121 .583,27.333
1122 .333,22.500
```

```
1128 1
1010 139,99 651 0X000X
1020 3,1,0,0,1
1040 1
1050 3
1070 .2740,,2798,,0217,,666
1090 .8870,,6547,,398,3.72E7
1120 1.271,32.792
1121 1.271,29.292
1122 1.271,25.792
1123 .5833,21.833
1128 1
1010 141,99 671 0X0002
1020 3,1,0,0,1
1040 1
1050 3
1070 .2790,,2798,,0217,,666
1090 .3547,,3848,,629,3.72E7
1120 1.271,32.792
1121 1.271,29.292
1122 1.271,25.792
1123 .5833,21.833
1128 1
1010 143,99 101 0X000X
1020 3,1,0,0,1
1040 1
1050 3
1070 .2790,,2798,,0217,,666
1090 .8870,,6547,,398,3.72E7
1120 -3.667,33.667
1121 0.000,35.063
1122 3.667,33.667
1123 .583,27.333
1128 1
1010 145,99 123 0X0002
1020 3,2,0,0,1
1040 1
1050 3
1070 .2790,,2798,,0217,,666
1090 .3547,,3848,,629,3.72E7
1091 1.600,,4773,,500,3.72E7
1120 -3.667,33.667
1121 0.000,35.063
1122 3.667,33.667
1123 .583,27.333
1124 .333,22.500
1128 1
```

```
**********************************************************************

RBTRDATA                                          04/09/81     10:58 AM

**********************************************************************


1000
1010 33 3.0 .033 1.81E-9 8.6E-6 1.59
```

****

RWDATA1 04/16/81 11:28 AM

****

```
1000 999 99 10 6 3 4 3 0 0 1
1010 1 .0231 75.0  8130.
1020 2 2.52 45.0  21.0 1.11
1030  50. .385 .064 .896 3.0 2.3 1 .750 .515 .880 .780 2.0
```

This file contains a set of underground cable parameters and is read by main program UGZYGES1.

```
41 3 4.41E-5 -6.2E-4 1.43E-5 1.02E-5
42 3 1.88E-4 -2.97E-5 -1.23E-5 6.89E-5
43 3 8.58E-5 -1.61E-04 -8.46E-7 3.3E-5
44 3 8.11E-5 -9.86E-5 -1.14E-5 2.6E-5
45 3 6.4E-5 -7.55E-5 -9.51E-6 2.01E-5
46 3 4.84E-5 -5.06E-5 -8.52E-6 1.4E-5
47 3 3.38E-5 -3.04E-5 -6.92E-6 8.6E-6
48 3 2.01E-5 -2.03E-5 -3.68E-6 5.64E-6
49 3 2.42E-5 1.55E-1 2.42E-5 -7.84E-2
51 3 2.46E-5 7.71E-2 2.46E-5 -3.94E-2
55 3 2.41E-5 3.11E-1 2.41E-5 -1.56E-1
```

This file illustrates an input file for main program AFPRYGEN.

```
1000
1010
1020   41     3     0     3     0     0
1030
1040    4.41E-05
1050   -6.20E-04
1060
1070    1.43E-05     4.41E-05
1080    1.02E-05    -6.20E-04
1090
1100    1.43E-05     1.43E-05     4.41E-05
1110    1.02E-05     1.02E-05    -6.20E-04
1120
1130   42     3     0     3     0     0
1140
1150    1.88E-04
1160   -2.97E-05
1170
1180   -1.23E-05     1.88E-04
1190    6.89E-05    -2.97E-05
1200
1210   -1.23E-05    -1.23E-05     1.88E-04
1220    6.89E-05     6.89E-05    -2.97E-05
1230
1240   43     3     0     3     0     0
1250
1260    8.58E-05
1270   -1.61E-04
1280
1290   -8.46E-07     8.58E-05
1300    3.30E-05    -1.61E-04
1310
1320   -8.46E-07    -8.46E-07     8.58E-05
1330    3.30E-05     3.30E-05    -1.61E-04
1340
1350   44     3     0     3     0     0
1360
1370    8.11E-05
1380   -9.86E-05
1390
1400   -1.14E-05     8.11E-05
1410    2.60E-05    -9.86E-05
1420
1430   -1.14E-05    -1.14E-05     8.11E-05
1440    2.60E-05     2.60E-05    -9.86E-05
1450
1460   45     3     0     3     0     0
1470
1480    6.40E-05
```

This file represents a file generated by main program AFPRYGEN using file R1 as input.

```
1490   -7.55E-05
1500
1510   -9.51E-06    6.40E-05
1520    2.01E-05   -7.55E-05
1530
1540   -9.51E-06   -9.51E-06    6.40E-05
1550    2.01E-05    2.01E-05   -7.55E-05
1560
1570   46    3     0     3     0     0
1580
1590    4.84E-05
1600   -5.06E-05
1610
1620   -8.52E-06    4.84E-05
1630    1.40E-05   -5.06E-05
1640
1650   -8.52E-06   -8.52E-06    4.84E-05
1660    1.40E-05    1.40E-05   -5.06E-05
1670
1680   47    3     0     3     0     0
1690
1700    3.38E-05
1710   -3.04E-05
1720
1730   -6.92E-06    3.38E-05
1740    8.60E-06   -3.04E-05
1750
1760   -6.92E-06   -6.92E-06    3.38E-05
1770    8.60E-06    8.60E-06   -3.04E-05
1780
1790   48    3     0     3     0     0
1800
1810    2.01E-05
1820   -2.03E-05
1830
1840   -3.68E-06    2.01E-05
1850    5.64E-06   -2.03E-05
1860
1870   -3.68E-06   -3.68E-06    2.01E-05
1880    5.64E-06    5.64E-06   -2.03E-05
1890
1900   49    3     0     3     0     0
1910
1920    2.42E-05
1930    1.55E-01
1940
1950    2.42E-05    2.42E-05
1960   -7.84E-02    1.55E-01
1970
1980    2.42E-05    2.42E-05    2.42E-05
1990   -7.84E-02   -7.84E-02    1.55E-01
2000
```

```
2010   51     3     0     3     0     0
2020
2030    2.46E-05
2040    7.71E-02
2050
2060    2.46E-05     2.46E-05
2070   -3.94E-02     7.71E-02
2080
2090    2.46E-05     2.46E-05     2.46E-05
2100   -3.94E-02    -3.94E-02     7.71E-02
2110
2120   55     3     0     3     0     0
2130
2140    2.41E-05
2150    3.11E-01
2160
2170    2.41E-05     2.41E-05
2180   -1.56E-01     3.11E-01
2190
2200    2.41E-05     2.41E-05     2.41E-05
2210   -1.56E-01    -1.56E-01     3.11E-01
```

TRANFILE 04/09/81 10:58 AM

5	5	-91.417	-26.978	5.130
5	10	-86.370	-46.531	5.130
5	15	-82.190	-42.069	5.130
5	25	-78.699	-47.761	5.130
5	40	-76.133	-51.067	5.130
5	50	-73.612	-53.817	5.130
5	75	-70.341	-57.353	5.130
5	100	-70.963	-66.628	5.130
5	167	-64.488	-63.162	5.130
86	5	-93.092	-17.986	6.990
86	10	-88.618	-45.353	6.990
86	15	-84.211	-39.867	6.990
86	25	-81.137	-46.397	6.990
86	40	-78.915	-49.721	6.990
86	50	-76.294	-53.735	6.990
86	75	-72.926	-56.490	6.990
86	100	-74.251	-67.525	6.990
86	167	-67.154	-65.166	6.990
88	5	-94.056	-10.045	8.130
88	10	-90.145	-42.953	8.130
88	15	-85.560	-36.856	8.130
88	25	-82.810	-43.641	8.130
88	40	-80.844	-46.641	8.130
88	50	-78.121	-51.934	8.130
88	75	-74.636	-57.861	8.130
88	100	-76.405	-66.632	8.130
88	167	-68.850	-65.322	8.130
89	5	-94.965	2.265	9.510
89	10	-92.165	-37.595	9.510
89	15	-87.360	-30.795	9.510
89	25	-85.043	-37.346	9.510
89	40	-83.468	-39.169	9.510
89	50	-80.603	-47.125	9.510
89	75	-76.914	-55.398	9.510
89	100	-79.350	-63.521	9.510
89	167	-71.053	-64.357	9.510
13	5	-95.189	28.688	12.010
13	10	-96.022	-15.992	12.010
13	15	-90.514	-9.240	12.010
13	25	-89.195	-10.675	12.010
13	40	-88.169	-3.967	12.010
13	50	-85.831	-22.746	12.010
13	75	-81.946	-41.986	12.010
13	100	-86.561	-42.658	12.010
13	167	-75.858	-57.474	12.010
15	5	-93.230	54.014	15.030
15	10	-97.073	33.459	15.030
15	15	-91.173	33.435	15.030
15	25	-86.962	43.442	15.030

This file is read by subroutine TRANAD which is called by NTWKERS4.

15	40	-86.004	54.657	15.030
15	50	-86.411	43.220	15.030
15	75	-86.805	17.637	15.030
15	100	-87.605	52.411	15.030
15	167	-83.760	-16.248	15.030
18	5	-91.563	64.110	17.010
18	10	-94.787	56.572	17.010
18	15	-89.299	54.399	17.010
18	25	-86.242	63.192	17.010
18	40	-82.884	70.038	17.010
18	50	-83.002	65.486	17.010
18	75	-83.741	55.762	17.010
18	100	-82.637	72.969	17.010
18	167	-82.863	43.798	17.010
20	5	-89.296	73.235	20.010
20	10	-91.286	72.536	20.010
20	15	-86.162	70.576	20.010
20	25	-82.718	75.987	20.010
20	40	-79.391	79.474	20.010
20	50	-79.031	77.900	20.010
20	75	-78.903	75.133	20.010
20	100	-78.139	81.788	20.010
20	167	-76.536	74.024	20.010
22	5	-87.994	76.921	21.990
22	10	-89.462	77.541	21.990
22	15	-84.435	76.002	21.990
22	25	-80.947	79.986	21.990
22	40	-77.685	82.423	21.990
22	50	-77.143	81.487	21.990
22	75	-76.681	79.965	21.990
22	100	-76.167	84.140	21.990
22	167	-73.827	79.849	21.990
25	5	-86.302	80.597	24.990
25	10	-87.271	81.810	24.990
25	15	-82.307	80.766	24.990
25	25	-78.629	83.421	24.990
25	40	-75.652	84.972	24.990
25	50	-74.940	84.464	24.990
25	75	-74.183	83.698	24.990
25	100	-73.928	86.057	24.990
25	167	-70.951	83.909	24.990

ZYDA1164 04/09/81 11:06 AM

```
1010  101   99   11     6     4     4     3     0     0     1    5.0100000E 00
1020  4.46730893E-03   6.00693184E-02
1030  4.02752461E-03   2.85564090E-02
1040  4.04100039E-03   2.45977675E-02
1050  4.11022425E-03   2.42747199E-02
1060  4.02752461E-03   2.85564090E-02
1070  4.43603484E-03   6.01121765E-02
1080  4.02752461E-03   2.85564090E-02
1090  4.09554242E-03   2.41744465E-02
1100  4.04100039E-03   2.45977675E-02
1110  4.02752461E-03   2.85564090E-02
1120  4.46730893E-03   6.00693184E-02
1130  4.11047641E-03   2.47324028E-02
1140  4.11022425E-03   2.42747199E-02
1150  4.09554242E-03   2.41744465E-02
1160  4.11047641E-03   2.47324028E-02
1170  5.52620267E-03   6.65014805E-02
1180  0.               2.78221602E-07
1190  0.              -7.66795863E-08
1200  0.              -4.01249167E-08
1210  0.              -4.11331311E-08
1220  0.              -7.66795863E-08
1230  0.               2.91851169E-07
1240  0.              -7.64225394E-08
1250  0.              -3.50210567E-08
1260  0.              -4.01249172E-08
1270  0.              -7.64225394E-08
1280  0.               2.79337403E-07
1290  0.              -4.43645352E-08
1300  0.              -4.11331311E-08
1310  0.              -3.50210567E-08
1320  0.              -4.43645347E-08
1330  0.               2.47604891E-07
9999
1010  103   99   11     6     4     4     3     0     0     1    5.0100000E 00
1020  5.13337535E-03   6.46738438E-02
1030  4.02752461E-03   2.85564090E-02
1040  4.04100039E-03   2.45977675E-02
1050  4.11022425E-03   2.42747199E-02
1060  4.02752461E-03   2.85564090E-02
1070  5.10410132E-03   6.47167024E-02
1080  4.02752461E-03   2.85564090E-02
1090  4.09554242E-03   2.41744465E-02
1100  4.04100039E-03   2.45977675E-02
1110  4.02752461E-03   2.85564090E-02
1120  5.13337535E-03   6.46738438E-02
1130  4.11047641E-03   2.47324028E-02
1140  4.11022425E-03   2.42747199E-02
1150  4.09554242E-03   2.41744465E-02
```

This file illustrates the output of proprietary program DISEM7S3 and represents matrix Z-Y data for overhead lines. Files of this format are read by main program FEEDPUS5.

```
1160  4.11047641E-03  2.47324028E-02
1170  5.52620267E-03  6.65014805E-02
1180  0.               2.66748273E-07
1190  0.              -7.14466637E-08
1200  0.              -3.78893779E-08
1210  0.              -4.01741818E-08
1220  0.              -7.14466637E-08
1230  0.               2.78815033E-07
1240  0.              -7.10157977E-08
1250  0.              -3.45019719E-08
1260  0.              -3.78893779E-08
1270  0.              -7.10157977E-08
1280  0.               2.67790184E-07
1290  0.              -4.32578800E-08
1300  0.              -4.01741818E-08
1310  0.              -3.45019715E-08
1320  0.              -4.32578800E-08
1330  0.               2.46929737E-07
9999
1010  105   99   11    6    4    4    3    0    0    1   5.0100000E 00
1020  5.60063706E-03  6.70498572E-02
1030  4.02752461E-03  2.85564090E-02
1040  4.04100039E-03  2.45977675E-02
1050  4.11022425E-03  2.42747199E-02
1060  4.02752461E-03  2.85564090E-02
1070  5.57136303E-03  6.70927148E-02
1080  4.02752461E-03  2.85564090E-02
1090  4.09554242E-03  2.41744465E-02
1100  4.04100039E-03  2.45977675E-02
1110  4.02752461E-03  2.85564090E-02
1120  5.60063706E-03  6.70498572E-02
1130  4.11047641E-03  2.47324028E-02
1140  4.11022425E-03  2.42747199E-02
1150  4.09554242E-03  2.41744465E-02
1160  4.11047641E-03  2.47324028E-02
1170  5.52620267E-03  6.65014805E-02
1180  0.               2.56910280E-07
1190  0.              -6.69071749E-08
1200  0.              -3.59640877E-08
1210  0.              -3.93234920E-08
1220  0.              -6.69071749E-08
1230  0.               2.67728581E-07
1240  0.              -6.64987701E-08
1250  0.              -3.40181141E-08
1260  0.              -3.59640877E-08
1270  0.              -6.64987701E-08
1280  0.               2.57889980E-07
1290  0.              -4.22807638E-08
1300  0.              -3.93234912E-08
1310  0.              -3.40181141E-08
1320  0.              -4.22807638E-08
1330  0.               2.46326014E-07
```

```
9999
1010  107   99   11     6     4     4     3     0     0     1    5.0100000E 00
1020  4.30641357E-03  5.86463432E-02
1030  4.02752461E-03  2.85564090E-02
1040  4.04100039E-03  2.45977675E-02
1050  4.11022425E-03  2.42747199E-02
1060  4.02752461E-03  2.85564090E-02
1070  4.35913954E-03  5.86692012E-02
1080  4.02752461E-03  2.85564090E-02
1090  4.09554242E-03  2.41744465E-02
1100  4.04100039E-03  2.45977675E-02
1110  4.02752461E-03  2.85564090E-02
1120  4.30641357E-03  5.86463432E-02
1130  4.11047641E-03  2.47324028E-02
1140  4.11022425E-03  2.42747199E-02
1150  4.09554242E-03  2.41744465E-02
1160  4.11047641E-03  2.47324028E-02
1170  5.52620267E-03  6.65014805E-02
1180  0.               2.88433131E-07
1190  0.              -8.16380110E-08
1200  0.              -4.21016915E-08
1210  0.              -4.19583785E-08
1220  0.              -8.18380101E-08
1230  0.               3.03552493E-07
1240  0.              -6.13577161E-08
1250  0.              -3.54440908E-08
1260  0.              -4.21016915E-08
1270  0.              -8.13577161E-08
1280  0.               2.89615841E-07
1290  0.              -4.53214213E-08
1300  0.              -4.19583790E-08
1310  0.              -3.54440903E-08
1320  0.              -4.53214213E-08
1330  0.               2.48181017E-07
9999
1010  109   99   11     6     4     4     3     0     0     1    5.0100000E 00
1020  4.46730893E-03  6.00693184E-02
1030  4.02752461E-03  2.85564090E-02
1040  4.04100039E-03  2.45977675E-02
1050  4.11022425E-03  2.42747199E-02
1060  4.02752461E-03  2.85564090E-02
1070  4.43803484E-03  6.01121765E-02
1080  4.02752461E-03  2.85564090E-02
1090  4.09554242E-03  2.41744465E-02
1100  4.04100039E-03  2.45977675E-02
1110  4.02752461E-03  2.85564090E-02
1120  4.46730893E-03  6.00693184E-02
1130  4.11047641E-03  2.47324028E-02
1140  4.11022425E-03  2.42747199E-02
1150  4.09554242E-03  2.41744465E-02
1160  4.11047641E-03  2.47324028E-02
1170  6.43932709E-03  6.77828332E-02
```

```
1180   0.          2.77P00072E-07
1190   0.         -7.72384805E-08
1200   0.         -4.05795624E-08
1210   0.         -3.85956866E-08
1220   0.         -7.72384805E-08
1230   0.          2.91545603E-07
1240   0.         -7.680962H0E-08
1250   0.         -3.28606577E-08
1260   0.         -4.05795633E-08
1270   0.         -7.680962H0E-08
1280   0.          2.78847040E-07
1290   0.         -4.16277501E-08
1300   0.         -3.85956866E-08
1310   0.         -3.28606577E-08
1320   0.         -4.16277497E-08
1330   0.          2.32330498E-07
9999
```